



Escuela
Politécnica
Superior

Sistema de realidad virtual y visualización 3D del cuerpo humano en clínicas dietéticas



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

Luis Fernando Pérez Pérez

Tutor/es:

Jorge Azorín López

Andrés Fuster Guilló



Universitat d'Alacant
Universidad de Alicante

Junio 2019

UNIVERSIDAD DE ALICANTE

Departamento de Tecnología Informática y Computación

Trabajo de Fin de Grado

Sistema de realidad virtual y visualización 3D del cuerpo humano en clínicas dietéticas

Luis Fernando Pérez Pérez

Tutores

Dr. Jorge Azorín López

Dr. Andrés Fuster Guilló

Memoria presentada para aspirar al grado de:

GRADUADO EN INGENIERÍA INFORMÁTICA

Alicante, 15 de junio de 2019

“No me diga que el cielo es el límite cuando hay huellas en la luna.”

Paul Brandt

AGRADECIMIENTOS

Es difícil acordarse de todas las personas que merecerían un agradecimiento, ya que han sido 4 años de cientos de momentos y mi mente no es capaz de recordarlos todos. Por ello, en estos párrafos agradeceré a todas aquellas personas que más me han marcado en este largo viaje.

En primer lugar, tengo que realizar un agradecimiento especial a mis padres. No sólo por el apoyo que me habéis dado a lo largo de la carrera, sino por el apoyo que me habéis dado durante toda mi vida, por la educación y los valores que me habéis transmitido y que, en gran medida, me hacen ser la persona que soy hoy. No existe suficiente tiempo en el universo ni palabras en el diccionario para poder agradecerlos todo lo que habéis hecho por mí, os quiero.

A mi pareja Raquel, mi apoyo más constante, la que ha soportado mis momentos de agobio, mis insufribles charlas sobre lo que hacía, sin ti esta etapa no hubiera sido lo mismo. Gracias por aguantarme, gracias por apoyarme, gracias por cuidarme, pero, sobre todo, gracias por quererme como yo te quiero a ti.

A mis hermanos por cada momento que os habéis preocupado por mí, por vuestros consejos, por vuestro apoyo presente, pasado y futuro y las largas charlas y comidas en casa. Junto a nuestros padres, habéis sido y sois una gran influencia en mi vida.

A mis sobrinos, aunque algunos aún son pequeños, han sido una parte muy importante en mi vida y espero que algún día cuando crezcan pueda ser para ellos un referente en el que fijarse.

A Ian y Pedro, mis dos mejores amigos, porque durante esta etapa he tenido que aislarme para centrarme en luchar por lo que me apasiona y les he dejado un poco de lado, pero pese a ello siguen apoyándome.

A mis compañeros de carrera, por haber vivido este largo camino conmigo y sobre todo a Juan Miguel y Dani, dos personas que me llevo para toda la vida y que han sido un apoyo y un referente para mí en todos los aspectos.

A todos los profesores que me han impartido clase a lo largo de la carrera. Algunos más y algunos menos, pero todos me han aportado grandes enseñanzas y me han transmitido su pasión por este campo.

Por último, a todo el equipo de investigación que me ha dado su apoyo en este trabajo y de los que me llevo grandes aprendizajes.

RESUMEN

Los sistemas de visualización de modelos 3D realistas tienen gran utilidad en ámbitos como los videojuegos, la arquitectura, ingeniería entre otros. Uno de los ámbitos que empieza a tener mayores necesidades de estos sistemas es el sanitario. En este campo, tecnologías como la realidad virtual están teniendo una gran relevancia en la activación de diversas funciones cognitivas en enfermedades como el Alzheimer, el autismo o problemas psicológicos como las fobias.

Tech4Diet es un proyecto de investigación titulado “Modelado y visualización 4D del cuerpo humano para la mejora de la adherencia al tratamiento dietético nutricional de la obesidad” financiado por el ministerio de economía, industria y competitividad con referencia “TIN2017-89069-R” que se enmarca en la mejora del tratamiento de la obesidad desde un enfoque multidisciplinar. Aprovechando el potencial de áreas en pleno desarrollo como el modelado 3D o la realidad virtual aborda problemas abiertos como los aspectos psicológicos de la adherencia al tratamiento, abriendo oportunidades tanto en la intervención dietético-nutricional, como en la solución de nuevos retos tecnológicos.

Este trabajo forma parte de dicho proyecto. En concreto, se ha desarrollado un sistema de visualización que permite tanto a especialistas médicos como a pacientes con obesidad visualizar y analizar el progreso de adelgazamiento a lo largo del tiempo. Este sistema de visualización tiene un módulo de visualización 3D y análisis orientado al especialista médico y otro módulo de realidad virtual orientado al paciente.

El módulo de visualización 3D se ha desarrollado como una aplicación de escritorio destinada al especialista médico que le permite visualizar los modelos 3D escaneados de los pacientes y tener acceso a los datos registrados de cada una de las sesiones permitiéndole realizar análisis comparativos entre diferentes sesiones de un mismo paciente para observar en que zonas ha perdido volumen a lo largo del tiempo. Una

funcionalidad muy importante en esta aplicación ha sido la incorporación de medidas reales sobre el cuerpo digital escaneado. Para resolver este aspecto se han abordado tres aproximaciones y se ha realizado una comparativa de ellas para observar la precisión de cada una.

El módulo de realidad virtual se ha desarrollado como una aplicación móvil destinada al paciente que permite visualizar los modelos 3D del cuerpo humano mediante realidad virtual con el objetivo de incrementar su adherencia al tratamiento.

ÍNDICE GENERAL

1	Introducción	1
1.1	Motivación y contexto.....	1
1.2	Estado del arte	5
1.3	Objetivos	8
1.4	Planificación	10
1.4.1	Metodología.....	11
1.4.2	Forma de organización	12
2	Tecnologías software y hardware.....	17
2.1	Unity.....	17
2.2	Visual Studio.....	20
2.3	MySQL.....	21
2.4	Lenguajes de programación (C#, C++)	23
2.5	Balsamiq Mockups.....	24
2.6	Bitbucket.....	25
2.7	SDK de Android	26
2.8	Dispositivo móvil	26
2.9	Lentes de realidad virtual.....	27
3	Sistema de ayuda al diagnóstico y tratamiento de la obesidad.....	29
3.1	Requisitos	30
3.1.1	Requisitos funcionales	30
3.1.2	Requisitos no funcionales	31
3.2	Diseño de interfaz gráfica.....	32
3.2.1	Mockups.....	33

3.2.2	Diagramas de pantalla.....	34
3.2.3	Implementación de la interfaz.....	34
3.2.4	Problemas encontrados.....	36
3.3	Lógica de la aplicación.....	37
3.3.1	Diseño de diagrama de clase.....	38
3.3.2	Implementación de la lógica.....	39
3.4	Base de datos.....	43
3.4.1	Especificación de datos.....	45
3.4.2	Diseño de la base de datos.....	48
3.4.3	Implementación lógica de negocio.....	49
3.4.4	Problemas encontrados.....	49
3.5	Comunicación aplicación – servidor.....	50
3.5.1	Análisis de requisitos.....	51
3.5.2	Diseño protocolo de comunicación.....	51
3.5.3	Implementación.....	54
4	Obtención de medidas antropométricas.....	55
4.1	Aproximación mediante la búsqueda del vértice más cercano.....	56
4.1.1	Problemas encontrados.....	59
4.2	Aproximación mediante el método de Montecarlo.....	60
4.2.1	Problemas encontrados.....	62
4.3	Aproximación mediante simulación de rayos.....	63
4.3.1	Simulación de rayos desde el exterior.....	64
4.4	Experimentación.....	66
5	Sistema de realidad virtual.....	69
5.1	Requisitos.....	70
5.1.1	Requisitos funcionales.....	70
5.1.2	Requisitos no funcionales.....	70
5.2	Implementación.....	70

5.2.1	Transmisión del movimiento de cabeza al objeto 3D.....	71
5.2.2	Bloqueo de la cámara en realidad virtual.....	72
5.2.3	Sincronización con aplicación de escritorio.....	73
6	Conclusión.....	77
6.1	Conclusiones	77
6.2	Líneas futuras	78
6.3	Conclusiones personales.....	79
7	Referencias	81
8	Anexos	85
8.1	Anexo I Bocetos del sistema de visualización.....	85
8.2	Anexo II Interfaz gráfica	88

ÍNDICE DE FIGURAS

Figura 1 Proceso del proyecto Tech4Diet	2
Figura 2 Cabina de adquisición. Diseño (Izquierda) y resultado (Derecha)	3
Figura 3 Muestra visualización sistema 2D.....	4
Figura 4 Muestra sistema realidad virtual	4
Figura 5 Muestras de software de empresas privadas.....	7
Figura 6 Diagrama Gantt del proyecto.....	10
Figura 7 Esquema de SCRUM (scrum.org).....	11
Figura 8 Tablero fase de visualización	13
Figura 9 Tablero unificado del proyecto	13
Figura 10 Leyenda de colores utilizada en Trello	14
Figura 11 Extracto del Excel de las últimas semanas.....	14
Figura 12 Logotipo de Unity.....	17
Figura 13 Muestra del interfaz gráfico de Unity y Unreal Engine.....	18
Figura 14 Logotipo de Visual Studio	20
Figura 15 Muestra del editor de Visual Studio	21
Figura 16 Logotipo de MySQL.....	21
Figura 17 Logotipo Balsamiq	24
Figura 18 Muestra de la herramienta Balsamiq Mockups.....	24
Figura 19 Logotipo Bitbucket	25
Figura 20 Muestra del interfaz web de BitBucket	25
Figura 21 Logotipo del SDK de Android	26
Figura 22 Diagrama del protocolo seguido.....	30
Figura 23 Proceso para elaborar el diseño gráfico	32
Figura 24 Diagrama de pantallas del sistema de visualización.....	34
Figura 25 Escena vista desde el exterior para abarcar el plano visto desde la cámara....	36

Figura 26 Proceso para elaborar la lógica de la aplicación	37
Figura 27 Diagrama de clases del sistema	38
Figura 28 Esquema del fov	39
Figura 29 Resultados del fov. Izquierda fov por defecto y derecha fov reducido	40
Figura 30 Resultado rotación. A la izquierda sin rotar, a la derecha rotado	41
Figura 31 Factor temporal. Izquierda cuerpo 1, derecha cuerpo 2	42
Figura 32 Comparativa entre dos cuerpos	42
Figura 33 Esquema MVC	44
Figura 34 Proceso para elaborar la base de datos	44
Figura 35 Ejemplo visual del IMC en mujeres	46
Figura 36 Ejemplo visual del IMC en hombres	46
Figura 37 Diagrama de la base de datos	48
Figura 38 Esquema de los sistemas que hay que comunicar	50
Figura 39 Proceso para elaborar la comunicación aplicación - servidor	51
Figura 40 Arquitectura de comunicación	52
Figura 41 Esquema protocolo de comunicación	53
Figura 42 Armadillo man	56
Figura 43 Partes de una malla	56
Figura 44 Ejemplo de una malla triangular	57
Figura 45 Vértices seleccionados. Izquierda vista frontal, derecha vista sin la malla	58
Figura 46 Líneas formadas entre todos los puntos	59
Figura 47 Problema extremidades. Izquierda vista frontal, derecha vista sin la malla ...	59
Figura 48 Problema de la distancia. Izquierda resultado real, derecha resultado teórico	60
Figura 49 Plano cortando la malla	61
Figura 50 Puntos generados. Izquierda vista frontal, derecha vista sin la malla.	61
Figura 51 Detección sobre el colisionador de Unity	62
Figura 52 Tramo donde no detecta las colisiones correctamente	63
Figura 53 Vista del plano. Izquierda vista frontal, derecha vista frontal trasera	64

Figura 54 Circunferencia cortando la malla	65
Figura 55 Rayos generados. Izquierda vista frontal, derecha vista sin la malla.	65
Figura 56 Cuerpos usados en la experimentación. Modelos 1, 2, 3 y 4 respectivamente..	66
Figura 57 Proceso para elaborar el sistema de realidad virtual.....	69
Figura 58 Muestra de la transmisión de movimiento.....	71
Figura 59 Funcionamiento del bloqueo de cámara	72
Figura 60 Esquema de los sistemas que hay que comunicar	73
Figura 61 Pantalla de espera.....	74
Figura 62 Protocolo de comunicación.....	74
Figura 63 Sincronización. Izquierda aplicación escritorio, derecha aplicación VR.....	75
Figura 64 Mockup del menú principal	85
Figura 65 Mockup de búsqueda de un paciente	85
Figura 66 Mockup de la visualización del cuerpo del paciente.....	86
Figura 67 Mockup del formulario sobre el paciente.....	86
Figura 68 Mockup de la postura que debe adoptar el paciente.....	86
Figura 69 Mockup del progreso de escaneo.....	87
Figura 70 Mockup del resultado del escaneo.....	87
Figura 71 Menú principal.....	88
Figura 72 Cargar paciente.....	88
Figura 73 Escena especialista	89
Figura 74 Alta paciente	89
Figura 75 Asistente.....	90
Figura 76 Datos administrativos	90
Figura 77 Sesiones	91
Figura 78 Nueva sesión.....	91
Figura 79 Progreso escaneo.....	92
Figura 80 Fin escaneo.....	92

ÍNDICE DE TABLAS

Tabla 1 Comparativa entre Unity y Unreal	20
Tabla 2 Comparativa de SGBD.....	23
Tabla 3 Características Samsung Galaxy S8	27
Tabla 4 Características de VRBox.....	27
Tabla 5 Categorías del IMC	46
Tabla 6 Categorías de complexión	47
Tabla 7 Comparativa obtención de medidas	67

1 INTRODUCCIÓN

En este capítulo se expone la motivación y puesta en contexto del proyecto. Se presenta un estado del arte en el que se expone la situación actual y los desarrollos más avanzados para solventar problemáticas similares. Por último, se detallan los objetivos generales que conforman el proyecto y la planificación seguida para lograrlos.

1.1 Motivación y contexto

Hace 6 años, vi por primera vez una demo de unos cascos de realidad virtual, las Oculus Rift. Tenía 16 años y me impresionó como unos simples cascos podían transportarte a un mundo totalmente virtual, me parecía algo de película de ficción y nunca imaginaba que iba a poder ser yo 6 años después quién desarrollase un sistema basado en esta tecnología.

Cuando hace un año Andrés Fuster y Jorge Azorín me dieron la oportunidad de colaborar en este proyecto no podía rechazarlo. Este proyecto cumple dos de los requisitos que más valoro. Por un lado, me permite aportar ese granito de arena para mejorar el mundo ayudando a personas que sufren de obesidad, y es que esta enfermedad se ha incrementado a nivel mundial. En concreto, en España, según la Encuesta Nacional de Salud, la obesidad afecta al 17,4% de la población adulta mayor a 18 años y si se une el sobrepeso, la cifra se eleva a más de la mitad (54,5%). Este aumento se debe principalmente a una alimentación inadecuada, destacando el 44,4% de la población entre 1 y 15 años consume dulces a diario, y a una actividad sedentaria tanto en el mundo laboral, donde un 38% de la población pasan la mayor parte de su actividad principal sentados, como en el tiempo libre donde un 37,8% de la población

mayor a 15 años declara pasar este tiempo de manera sedentaria. Por otro lado, este proyecto me permite aprender a utilizar herramientas y tecnologías punteras como la realidad virtual de la mano de un equipo altamente cualificado como mis tutores Andrés Fuster Guilló, Jorge Azorín López u otros miembros como Marcelo Saval Calvo o Víctor Villena Martínez.

Este trabajo se enmarca en el proyecto con financiación nacional por parte del ministerio y la universidad de alicante Tech4Diet¹ bajo el título “Modelado y visualización 4D del cuerpo humano para la mejora de la adherencia al tratamiento dietético nutricional de la obesidad” con referencia “TIN2017-89069-R”. La finalidad del proyecto es aumentar la adherencia en tratamientos para la obesidad mediante tecnologías como la realidad virtual o la visualización de modelos 4D. Uno de los principales motivos del abandono de las dietas es por la falta de motivación. El individuo no percibe el cambio que sufre a lo largo del tratamiento por verse día tras día. Mediante el uso de herramientas tecnológicas se puede crear una base de datos de modelos del cuerpo en diferentes instantes de tiempo para permitir al paciente visualizar de forma inmersiva y realista la evolución de su cuerpo en el tiempo. Además, al tener el histórico de modelos, se pueden tomar medidas que antes no se tomaron y realizar comparaciones y análisis de la evolución de estas en los diferentes momentos del tratamiento.

El proceso seguido por este proyecto (Figura 1), está formado por diversas etapas. Estas etapas, a nivel técnico se pueden dividir en dos fases fundamentales, la fase de adquisición y la de visualización.

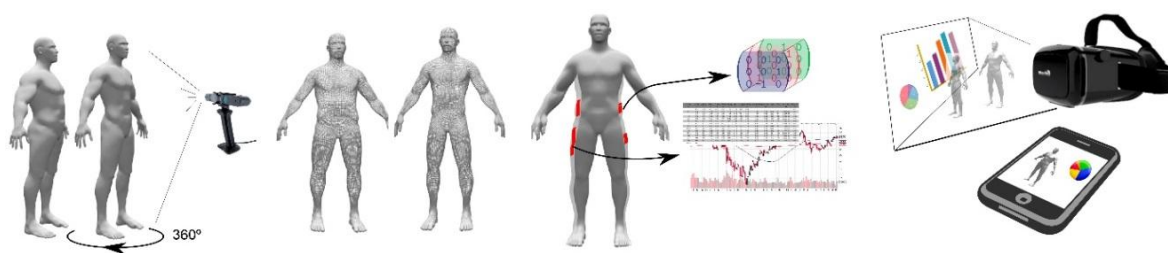


Figura 1 Proceso del proyecto Tech4Diet

¹ <http://tech4d.dtic.ua.es/>

- **Fase de adquisición:** En esta fase se realiza todo el proceso necesario para generar un modelo 3D del cuerpo humano del paciente. Para realizar esta fase ha sido necesario el montaje de una cabina formada por cuatro columnas de aluminio donde se sujetan 8 cámaras RGB-D Intel RealSense (Intel, 2015).



Figura 2 Cabina de adquisición. Diseño (Izquierda) y resultado (Derecha)

Una vez montada la cabina (Figura 2), se realizan diversas etapas hasta conseguir el modelo 3D. Primeramente, se realiza un calibrado tanto intrínseco como extrínseco de las cámaras. Víctor Villena realizó un análisis exhaustivo de diferentes métodos de calibrado para cámaras RGB-D (Villena Martínez, 2015). A continuación, se realiza la adquisición de las imágenes y las nubes de puntos del paciente desde cada cámara realizando una etapa de filtrado para cada nube de puntos con filtros como el statistical outlier removal (Rusu & Cousins, 2011) o el filtro bilateral (Tomasi & Manduchi, 2002). Se realiza una etapa posterior en la que se aplican las matrices de transformación calculadas en la etapa de calibrado a las nubes de puntos generando un registro con ellas. Por último, se realiza una etapa de mallado mediante algoritmos como el Poisson (Kazhdan, Bolitho, & Hoppe, 2006) y una etapa de texturizado proyectando las imágenes RGB

obtenidas de las cámaras. Esta fase no pertenece a este trabajo, pero si se aborda en el proyecto nacional.

- **Fase de visualización:** En esta fase se realiza un sistema para visualizar los modelos 3D generados en la fase de adquisición que permita la variación del punto de vista, la evolución temporal y la obtención de medidas precisas de dicho modelo. Este sistema será utilizado tanto por el especialista médico para permitirle la investigación y el análisis en el campo de la obesidad, cómo por el paciente para mejorar su adherencia al tratamiento. El sistema está compuesto por dos subsistemas:
 - Un sistema de visualización 2D de modelos 3D realistas contenido en una aplicación de escritorio (Figura 3).



Figura 3 Muestra visualización sistema 2D

- Un sistema de visualización en realidad virtual de modelos 3D realistas contenido en una aplicación móvil (Figura 4).



Figura 4 Muestra sistema realidad virtual

El reto que he decidido afrontar y en el que se centrará el presente trabajo es el desarrollo completo de la fase de visualización.

1.2 Estado del arte

En el presente estado del arte se van a abordar cuatro aspectos fundamentales en este proyecto. La situación actual existente en el ámbito sanitario y, concretamente, en la obesidad. El desarrollo de sistemas de visualización en el ámbito sanitario y su impacto. La obtención de medidas sobre modelos 3D y, finalmente, la situación actual de la tecnología de realidad virtual.

El sobrepeso y la obesidad se definen como una acumulación anormal o excesiva de grasa que resulta perjudicial para la salud. La prevalencia del sobrepeso y la obesidad se ha incrementado a nivel mundial, triplicándose durante las últimas tres décadas en los países de la Unión Europea (WHO, Organisation, 2006).

En la actualidad, la mayoría de los programas de intervención desarrollados para el tratamiento de los pacientes con sobrepeso y obesidad se han centrado en el establecimiento de indicaciones dietético-nutricionales y el fomento del ejercicio físico (Vieira et al., 2013), (Hall & Jordan, 2008). Aunque en términos generales, estos programas han demostrado ser eficaces para la pérdida de peso a corto plazo, su efectividad ha resultado limitada para el mantenimiento de peso a lo largo del tiempo. La adherencia al tratamiento ha sido considerada uno de los factores que afectan de forma determinante a la duración del tratamiento y a los resultados de las intervenciones con estos pacientes (WHO, 2003).

Dada la evidencia que demuestra que la falta de adherencia al tratamiento es una de las principales causas del fracaso de los programas de intervención, mejorar este componente en los pacientes con sobrepeso y obesidad contribuiría al resultado de los tratamientos y su mantenimiento a lo largo del tiempo produciendo disminución de costes en salud. Algunos autores han sugerido que las intervenciones nutricionales que refuerzan el seguimiento de las terapias mediante el uso de las tecnologías consiguen efectos beneficiosos a lo largo del tiempo (Roseman, Riddell, & Haynes, 2011). En una revisión de 15 estudios publicados de intervenciones realizadas en pacientes con problemas de sobrepeso y obesidad, se encontró que entre el 60-80% de los

participantes sometidos a la intervención, lograron mejoras significativas en la reducción del peso a lo largo del tiempo, cuando éstos eran reforzados mediante el uso de algún tipo de dispositivo multimedia interactivo (Ajie & Chapman-Novakofski, 2014).

Los resultados ponen de manifiesto cómo la falta de adherencia al tratamiento puede ser mejorada mediante la incorporación de técnicas basadas en la utilización de imágenes (2D) de la evolución del paciente en tratamiento dietético que potencian su experiencia cognitiva (Kuzmar, Rizo, & Cortés-Castell, 2014). Sin embargo, según mis investigaciones, no existen trabajos que aprovechen y cuantifiquen el potencial de las imágenes 3D realistas y las técnicas de realidad virtual para el refuerzo de la adherencia al tratamiento y el seguimiento en el tiempo de los pacientes con sobrepeso y obesidad.

El escaneado 3D del cuerpo humano fue desarrollado en gran medida para la industria textil (Phoebe R. Apeagyei, 2010). En la actualidad el modelado 3D del cuerpo humano está transformando nuestra capacidad de medirlo con precisión y visualizarlo, mostrando un gran potencial para aplicaciones sanitarias; desde la epidemiología, hasta el diagnóstico o la monitorización de pacientes (Treleaven & Wells, 2007). En epidemiología, por ejemplo, se han utilizado mediciones del tamaño y la forma del cuerpo para prevenir la prevalencia creciente de la obesidad (Lin, Chiou, Weng, Tsai, & Liu, 2002). Se pueden destacar diversas empresas como *Fitness Diagnostic Lab*, *Styku*, *Prototech Solutions* o *body visualizer* cuyo software proporcionan funciones de visualización corporal y seguimiento para el mercado del fitness. La Figura 5 muestra los diferentes sistemas de visualización de dichas empresas privadas. Las imágenes superiores corresponden a las empresas *Fitness Diagnostic Lab* y *Styku* respectivamente, mientras que las inferiores corresponden a *Prototech Solutions* y *body visualizer* respectivamente.



Figura 5 Muestras de software de empresas privadas

Estas empresas orientan sus sistemas a personas del sector fitness que quieren mejorar su rendimiento. Sin embargo, ninguna está orientada a personas con obesidad.

Uno de los aspectos más relevantes del presente trabajo ha sido el cálculo de medidas sobre los modelos 3D. Este aspecto es objeto de multitud de investigaciones científicas. Artículos como “Volumeter: 3D human body parameters measurement with a single Kinect” (He, Ji, Zeng, & Zhang, 2018) muestran algunas aproximaciones de la realización de medidas con buenos resultados. Además, publicaciones recientes como “Automatic human body feature extraction and personal size measurement” (Xiaohui, Xiaoyu, Liwen, & Qing, 2018) muestran métodos novedosos para la realización automática de medidas extrayendo previamente las características de interés. Existen muchas investigaciones que obtienen medidas de manera precisa, pero no he encontrado aproximaciones aprovechando el uso de las herramientas proporcionadas en motores gráficos como Unity.

La realidad virtual es una de las tecnologías que ha experimentado un auge importante en los últimos años. Esta tecnología tiene la capacidad de simular entornos tridimensionales realistas con los que el usuario puede interaccionar utilizando HMDs (Head Mounted Displays), también conocidos como gafas de realidad virtual. Inicialmente estos dispositivos han tenido precios elevados dada su complejidad, sin embargo, desarrollos recientes como “Google Cardboard” han permitido que cualquier teléfono pueda ser convertido en un HMD con costes muy bajos. Actualmente existen numerosos trabajos que utilizan el potencial de la realidad virtual. Destaca la experiencia inmersiva que aporta la realidad virtual, mejorando la concentración en el proceso formativo (Hussein & Nätterdal, 2015). Cabe esperar que el uso de estas tecnologías pueda ser efectivo en la mejora de la adherencia al tratamiento de pacientes con sobrepeso.

La revisión del estado del arte me permite afirmar que existe urgencia en proporcionar soluciones al problema de la obesidad a nivel mundial. Los tratamientos clásicos han demostrado una efectividad limitada para resolver la cronicidad. El uso de tecnologías para la reconstrucción 3D del cuerpo humano está suficientemente maduro en diferentes campos de aplicación y se observa un auge en el uso de dispositivos de visión 3D de bajo coste basados en tecnología RGB-D. El campo de la realidad virtual también está evolucionando notablemente, encontrando sistemas de bajo coste y experiencias de éxito de aplicación en diferentes ámbitos. Sin embargo, no se encuentran trabajos que aborden el cambio de forma a lo largo del tiempo (4D), ni el uso de dispositivos de adquisición versátiles (RGB-D) y realidad virtual para el tratamiento de la obesidad. Estas conclusiones permiten afirmar que existen importantes retos tecnológicos en la investigación multidisciplinar de las aplicaciones de tecnologías de modelado 3D/4D del cuerpo humano y su representación mediante realidad virtual para el tratamiento de la obesidad mediante el uso de dispositivos de propósito general.

1.3 Objetivos

Antes de concretar los objetivos específicos de este trabajo, cabe recordar que el objetivo del proyecto de investigación en el que se enmarca es proporcionar un sistema

que permite estudiar la evolución del cuerpo humano usando técnicas de visión 3D/4D para mejorar los tratamientos de la obesidad y su adherencia a ellos.

En este trabajo, centrado en la fase de visualización del proyecto, se pueden diferenciar los siguientes objetivos y tareas:

- Desarrollo de un sistema de visualización realista de modelos 3D para los especialistas médicos. Este objetivo contiene las siguientes tareas
 - Estudiar diferentes herramientas para el desarrollo de los sistemas de visualización.
 - Captura de requisitos del sistema.
 - Desarrollo de una interfaz gráfica para el sistema.
 - Desarrollo de la lógica del sistema.
 - Implementación de una base de datos donde guardar los datos de los pacientes.
 - Establecimiento de un protocolo de comunicación entre el sistema y el servidor donde se escanean los cuerpos.

- Obtención de medidas precisas sobre los objetos 3D. Las tareas relacionadas con este objetivo son las siguientes.
 - Desarrollo y estudio de diferentes aproximaciones para conseguir medidas precisas.
 - Comparativa de todas las aproximaciones con las medidas reales para establecer la precisión de cada una de ellas.

1-INTRODUCCIÓN

- Desarrollo de un sistema de visualización realista de modelos 3D en realidad virtual para los pacientes de obesidad. Las tareas relacionadas con este objetivo son las siguientes.
 - Estudiar diferentes dispositivos móviles para ejecutar el sistema de realidad virtual.
 - Estudiar diversos modelos de gafas de realidad virtual para decidir el mejor en relación calidad/precio para el sistema de realidad virtual.
 - Desarrollo de un sistema de sincronización en tiempo real entre ambos sistemas de visualización.

1.4 Planificación

El desarrollo de este trabajo comenzó en febrero de 2018 alineado con el proyecto de investigación al que pertenece. Dicho proyecto tiene una duración de 3 años, pero este trabajo se ha planificado en el primer año del proyecto.

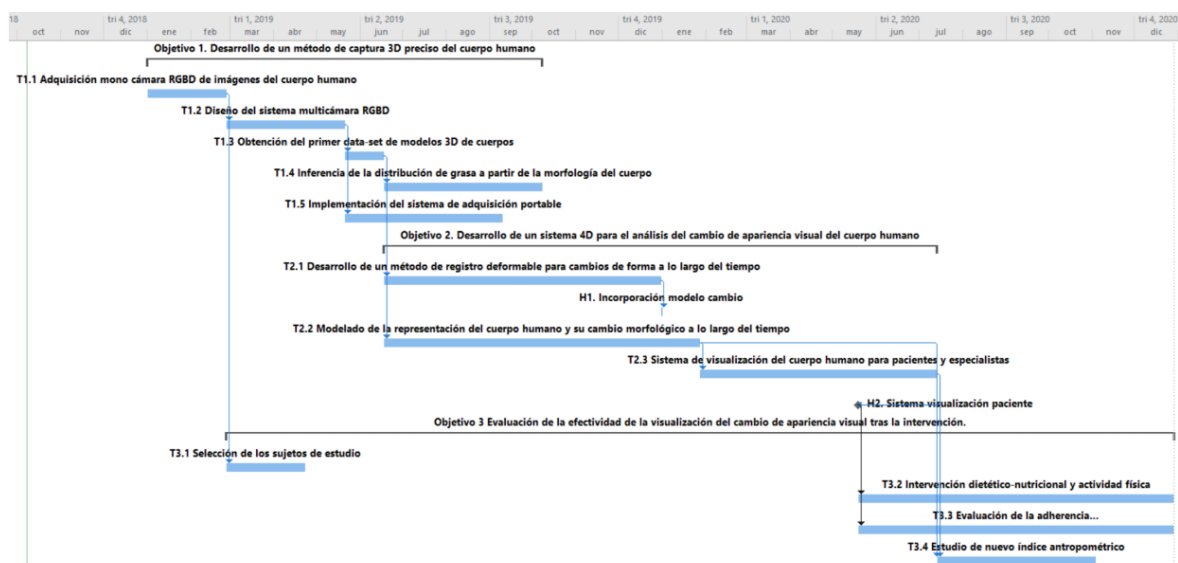


Figura 6 Diagrama Gantt del proyecto

Pese a que, en un inicio, se estimó que el sistema debería estar listo para junio de 2020 (Figura 6), la necesidad por parte del equipo sanitario de tener el sistema ha provocado adelantarlo un año antes, por lo que el objetivo es tener listo este sistema de visualización para junio de 2019, con la finalidad de que el equipo de nutrición y psicología, que colabora con este proyecto, pueda comenzar a utilizarlo para estudiar las consecuencias de este sistema en la adherencia del tratamiento en personas con obesidad.

1.4.1 Metodología

Para el desarrollo de este trabajo, hemos utilizado una metodología ágil llamada SCRUM (Schwaber & Sutherland, 2011). Esta metodología se caracteriza por ser iterativa incremental como se puede ver en su estructura (Figura 7).

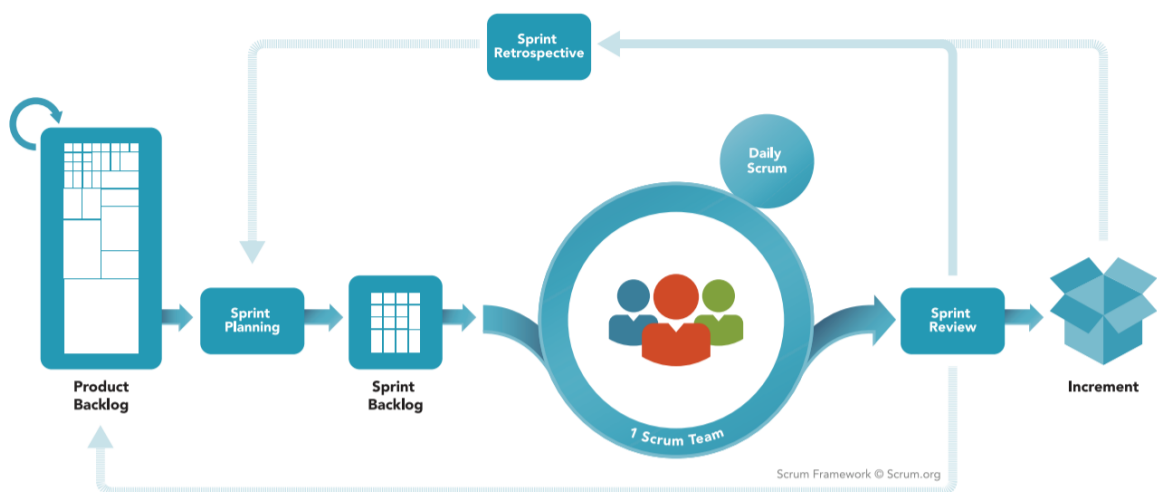


Figura 7 Esquema de SCRUM (scrum.org)

Los *Sprints* han sido semanales y nos reuníamos el equipo un día a la semana para realizar una retrospectiva tanto del trabajo realizado por cada uno de los miembros como de nuestra manera de trabajar.

Los motivos de utilizar esta metodología son variados. Por un lado, este proyecto está compuesto por un equipo relativamente pequeño que disponía de horarios muy

diversos para trabajar, por lo que una metodología como SCRUM nos ofrece una gran flexibilidad para compatibilizar los horarios. Por otro lado, los equipos de nutrición y psicología necesitaban poder ver los avances periódicamente así que una metodología iterativa incremental como es SCRUM nos permite ir mostrando resultados tangibles periódicamente. Por último, al ser un proyecto innovador, necesitábamos una metodología como SCRUM que nos ofrece una gran flexibilidad para cambiar distintos requisitos de manera ágil.

Un libro que recomiendo para obtener más información al respecto de esta metodología es “Guerreros Ágiles. Gestión de equipos SCRUM” (Fernando Llopis Pascual, 2015).

1.4.2 Forma de organización

La duración de este trabajo ha sido de 15 meses. Dado que este trabajo pertenece a un proyecto mayor, y pese a que el sistema de visualización lo he desarrollado de manera autónoma, ha sido necesario una coordinación con el equipo para cumplir los plazos establecidos. Para ello, nos reuníamos una vez a la semana para hablar de los avances realizados por cada miembro del equipo a lo largo de la semana y planificar las próximas tareas. Además, con el fin de conocer en todo momento que tareas estaba realizando cada uno, disponemos varios tableros de Trello donde podíamos saber que tareas realizaba cada uno y en qué estado se encontraban.

Al inicio del proyecto, la fase de visualización y la de adquisición disponían de tableros independientes ya que no existían dependencias entre ellas (Figura 8).

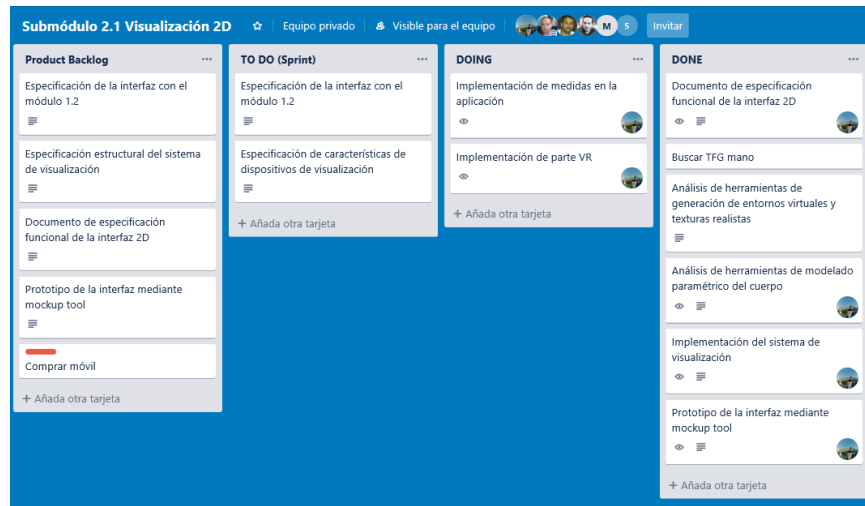


Figura 8 Tablero fase de visualización

Más adelante, cuándo el proyecto fue avanzando, decidimos unificar los tableros en uno ya que comenzaban a haber dependencias entre ambas fases (Figura 9).

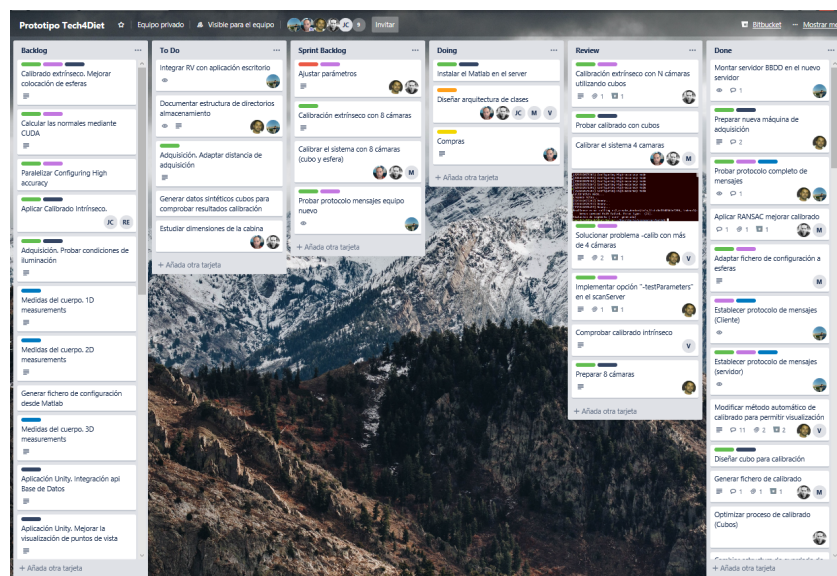


Figura 9 Tablero unificado del proyecto

Con la unificación de los tableros fue necesario añadir una leyenda de colores a cada tarea (Figura 10) para saber de manera visual en qué consistía aumentando así la agilidad del equipo.

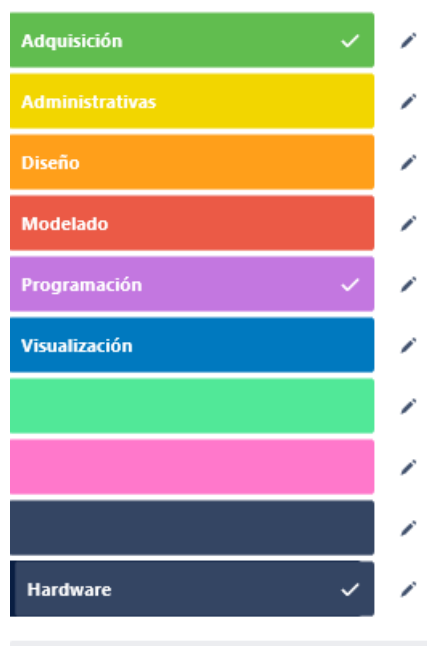


Figura 10 Leyenda de colores utilizada en Trello

Además, cada 2 meses aproximadamente nos reuníamos con el equipo de nutrición y psicología para mostrarles los avances conseguidos y darnos feedback para seguir con el proyecto.

Por último, a nivel personal me he organizado mediante un Excel donde apuntaba las tareas de cada semana y el tiempo que tenía para terminarlas (Figura 11). Además, en base a una leyenda de colores me mostraba de manera visual si quedaba mucho tiempo o no.

Trabajo	Entrega	Días Restantes		V	15
Conclusiones	12/05/2019	1		N	7
Repasar doc	15/05/2019	4		R	3
Preparar demo	17/05/2019	6			
Reunión final	20/05/2019	9			
Ensayo exposición	22/05/2019	11	●		

Figura 11 Extracto del Excel de las últimas semanas

2 TECNOLOGÍAS SOFTWARE Y HARDWARE

En este capítulo se detallan las diversas herramientas tecnológicas software y algunos componentes hardware que han sido necesarios en el desarrollo de este trabajo. Se justifica su uso y, en algunos casos, se detallan diversas comparativas para su selección.

2.1 Unity



Figura 12 Logotipo de Unity

Uno de los requisitos que había al desarrollar el sistema de visualización es que fuera realista. Es por este motivo que un motor de videojuegos como Unity podía ofrecer al sistema un punto mucho más realista.

Personalmente, hasta el inicio de este proyecto, nunca había utilizado un motor de videojuegos. Por ello, la decisión sobre que motor utilizar era muy importante para sacar el proyecto adelante.

En la actualidad, existen multitud de motores en el mercado. Sin embargo, hay dos motores que destacan por encima del resto, *Unity* y *Unreal Engine*.

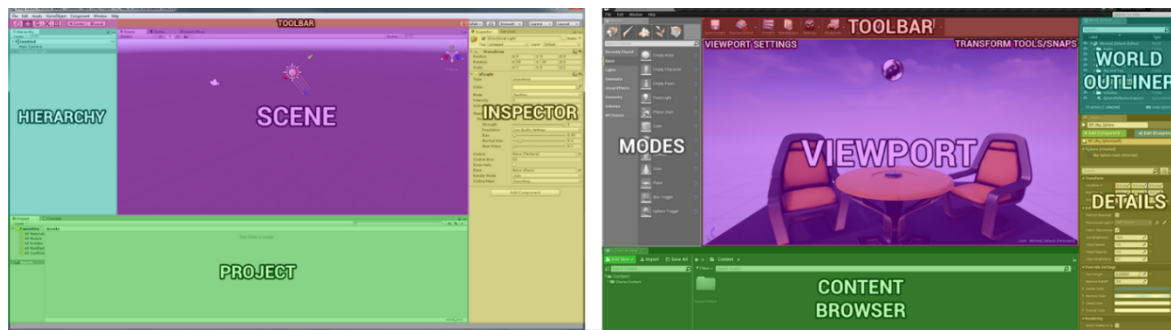


Figura 13 Muestra del interfaz gráfico de Unity y Unreal Engine

En cualquiera de los dos motores gráficos se podría haber implementado este sistema, pero decidí hacer un análisis comparativo de ambos para escoger el que mejor se adaptara a este trabajo.

Para realizar la comparativa, me basé en seis puntos clave para este trabajo. La documentación y la curva de aprendizaje por la inexperiencia y la necesidad de cumplir distintos plazos. El lenguaje en el que se puede desarrollar, ya que un lenguaje en el que estuviera familiarizado me permitiría aumentar la productividad. El aspecto económico debido a que el trabajo pertenece a un proyecto de investigación. Compatibilidad con realidad virtual y Android para evitar el uso de herramientas nuevas para realizar el sistema de realidad virtual en el móvil.

Las conclusiones obtenidas en cada aspecto clave son los siguientes:

- **Documentación:** Ambas herramientas poseen una gran cantidad de documentación, pero Unity destaca por tener una comunidad más activa y, sobre todo, por ofrecer de manera gratuita en su web un manual (Unity, 2018) extenso actualizado en castellano.
- **Curva de aprendizaje:** Aunque este aspecto es difícil de cuantificar de manera objetiva, la mayoría de los análisis señalan que Unity tiene una

menor curva de aprendizaje debido entre otras cosas a la extensa documentación oficial que tiene.

- **Lenguaje de desarrollo:** Unreal permite el desarrollo en C++ mientras que Unity lo hace en C# o JavaScript. Tanto C++ como C# son lenguajes en los que estoy familiarizado. Sin embargo, C# es un lenguaje de más alto nivel que C++ y, personalmente tengo mayor afinidad con C#.
- **Económico:** Ambas herramientas en un principio son gratuitas, ya que la filosofía que siguen ambas son similares. Su modelo de negocio se basa en cobrar una parte de los beneficios obtenidos con la aplicación desarrollada mediante su herramienta a partir de una cifra económica.

En el caso de Unity puedes utilizar la licencia personal que es gratuita y, en el caso de obtener un beneficio superior a 100.000\$ con el producto desarrollado en la herramienta, debes contratar la licencia PRO cuyo valor es de 1.500\$.

En el caso de Unreal es totalmente gratuita, pero si alcanzas un beneficio superior a 3.000\$ debes pagar el 5% de los beneficios obtenidos.

Por tanto, a nivel económico es más asequible Unity que Unreal.

- **Compatibilidad con realidad virtual:** Tanto Unity como Unreal cuentan con tecnología para el desarrollo de realidad virtual integrada en sus herramientas.
- **Compatibilidad con Android:** Ambas herramientas permiten el desarrollo en móviles Android.

Por último, en la Tabla 1 se muestra un resumen de la comparativa entre ambas herramientas.

	Unity	Unreal
Documentación	Muy amplia	Amplia
Curva de aprendizaje	Alta	Muy alta
Lenguaje de desarrollo	C# o JavaScript	C++
Económico	1500\$ a partir de 100.000\$ de beneficios	5% de las ganancias a partir de 3000\$ de beneficios
Compatibilidad con realidad virtual	Sí	Sí
Compatibilidad con Android	Sí	Sí

Tabla 1 Comparativa entre Unity y Unreal

2.2 Visual Studio



Figura 14 Logotipo de Visual Studio

Visual Studio es un entorno de desarrollo integrado (IDE) creado por Microsoft para sistemas operativos Windows. Este IDE admite lenguajes de desarrollo como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP.

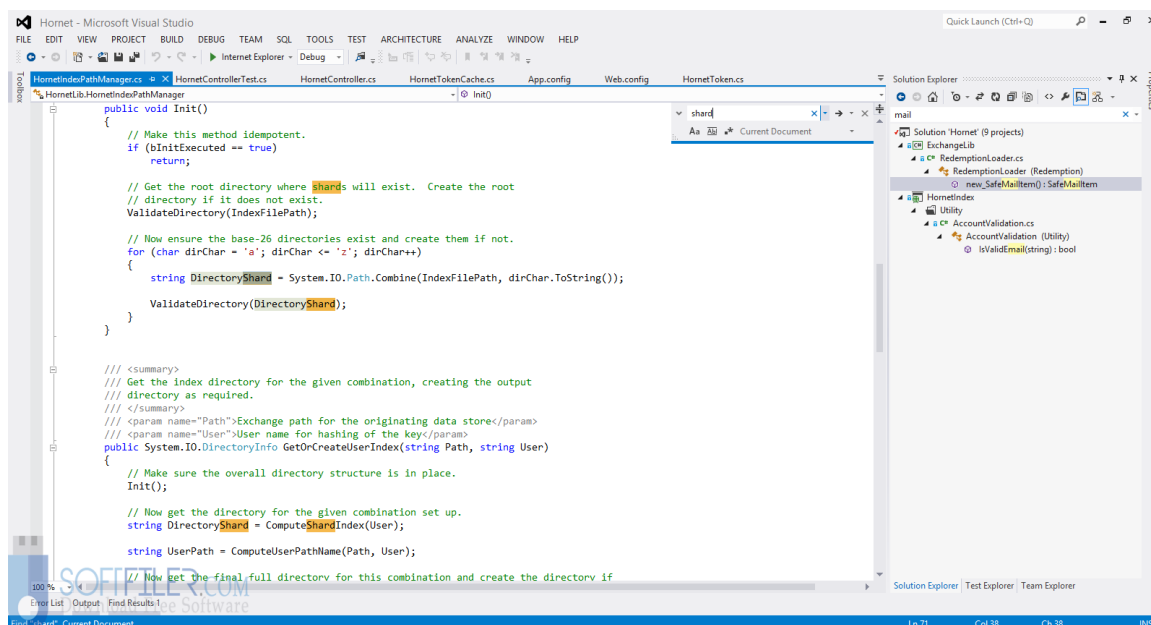


Figura 15 Muestra del editor de Visual Studio

La elección de este IDE se debe a dos motivos. Por un lado, Unity integra tanto Visual Studio como Mono para la programación de scripts y, por otro lado, ya he trabajado en varias ocasiones con este IDE por lo que estoy muy familiarizado con él.

2.3 MySQL



Figura 16 Logotipo de MySQL

En la actualidad, existen multitud de sistemas de gestión de bases de datos (SGBD por sus siglas). Es por esto, que ha sido necesario realizar un análisis de los más conocidos para decidir cuál era el que se adecuaba mejor a este proyecto. No obstante, inicialmente no se necesita una base de datos muy elaborada, por lo que cualquier SGBD podría cumplir su cometido en el trabajo.

Los SGBD que he decidido analizar son los siguientes:

- MySQL. Es el sistema gestor de bases de datos relacional por excelencia. Es un SGBD multihilo y multiusuario utilizado en la mayoría de las páginas de aplicaciones creadas como software libre y páginas web.
- MariaDB. Es una derivación de MySQL que cuenta con la mayoría de las características de este incluyendo varias extensiones. Nace a partir de la adquisición de MySQL por parte de Oracle para seguir la filosofía Open Source y es totalmente compatible con MySQL.
- SQLite. Es una biblioteca escrita en C que implementa un SGBD y que permite transacciones sin necesidad de un servidor ni configuraciones.
- Oracle. Ha sido el SGBD por excelencia para el mundo empresarial, considerado siempre como el más completo y robusto.

Por otro lado, las características que he tenido en cuenta a la hora de realizar el análisis son los siguientes:

- Familiaridad. La experiencia que he tenido con los distintos SGBD es importante tenerla en cuenta para mejorar mi productividad en esta etapa.
- Soporte. El soporte oficial o la actividad de la comunidad es otro aspecto muy importante para la selección del SGBD adecuado.
- Precio. El coste económico es un aspecto importante ya que la BBDD que hay que implementar no es demasiado compleja.
- Compatibilidad con C#. La compatibilidad con este lenguaje es importante, ya que se debe integrar con el sistema desarrollado en C#.

Aspectos como el rendimiento no se han tenido en cuenta debido a que no es una BBDD grande ni se van a realizar cientos de consultas simultáneas. La Tabla 2 muestra los resultados del análisis concluyendo que MySQL es el SGBD que mejor se adecua a este sistema.

	MySQL	MariaDB	SQLite	Oracle
Familiaridad	Muy Alta	Media	Muy baja	Baja
Soporte (Manual oficial y comunidad)	Manual/comunidad activa	No manual/comunidad activa	Manual/comunidad activa	Manual/comunidad activa
Precio	Gratis	Gratis	Gratis	Versión gratuita y de pago
Compatibilidad con C#	Si	No	Sí	Si

Tabla 2 Comparativa de SGBD

2.4 Lenguajes de programación (C#, C++)

En este trabajo se han empleado dos lenguajes de programación. Tanto el sistema de visualización de escritorio como de realidad virtual está desarrollado bajo C# debido a que es uno de los lenguajes base de Unity. Sin embargo, parte de la comunicación entre la aplicación de escritorio y el servidor se ha desarrollado en C++ ya que está integrado con la fase de adquisición desarrollada en dicho lenguaje.

2.5 Balsamiq Mockups



Figura 17 Logotipo Balsamiq

Balsamiq Mockups es la herramienta que he utilizado para realizar los bocetos de la interfaz gráfica de la aplicación de escritorio.

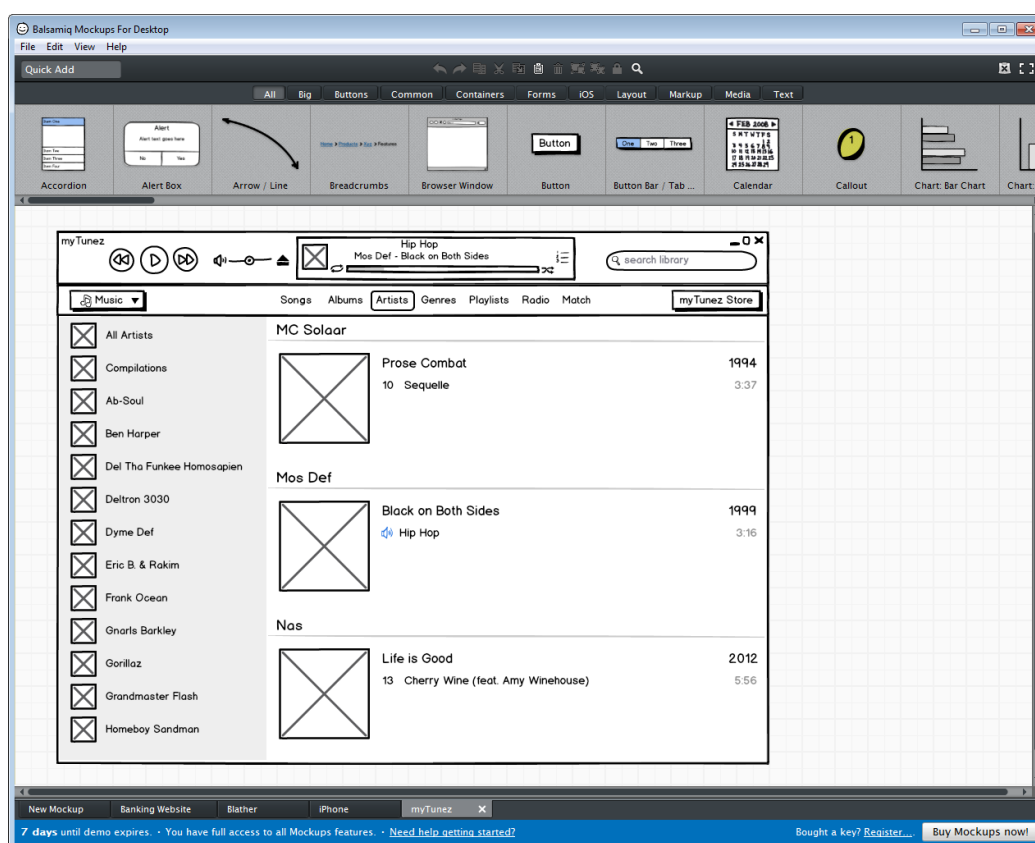


Figura 18 Muestra de la herramienta Balsamiq Mockups

Esta herramienta es muy intuitiva y sencilla de utilizar, además contiene una gran variedad de opciones para maquetar aplicaciones móviles, aplicaciones de escritorio, aplicaciones web entre otros. Esta herramienta, es de pago, pero permite un periodo de prueba de 30 días gratis.

2.6 Bitbucket



Figura 19 Logotipo Bitbucket

Bitbucket es un servicio de repositorios para proyectos que usan sistemas de control de versiones como Git o Mercurial.

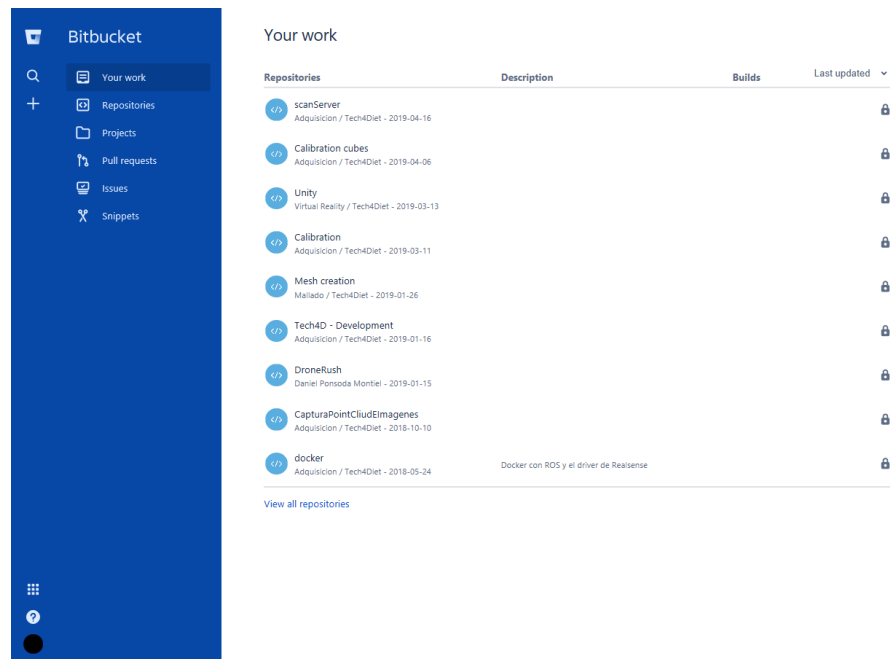


Figura 20 Muestra del interfaz web de BitBucket

La decisión de utilizar este repositorio se debe a que el equipo de investigación al que pertenezco ya lo estaba utilizando cuando inicié este trabajo. En esta herramienta he alojado dos repositorios. Uno para el sistema de visualización en escritorio y otro para el sistema de realidad virtual. Alojarlos ha servido para que mis tutores estuvieran al corriente en todo momento de los avances en la implementación.

2.7 SDK de Android



Figura 21 Logotipo del SDK de Android

El SDK (Software Development Kit) de Android, es un conjunto de herramientas de desarrollo que comprende un depurador de código, biblioteca, un simulador de teléfono basado en QEMU, documentación, ejemplos de código y tutoriales. Este SDK permite crear y depurar aplicaciones, así como controlar dispositivos Android que estén conectados (*SDK Tools / Android Developers*, n.d.). El uso del SDK ha sido necesario para el desarrollo del sistema de realidad virtual.

2.8 Dispositivo móvil

El dispositivo móvil seleccionado ha sido un Samsung Galaxy S8 dado que se ajustaba perfectamente al presupuesto y sus características lo hacen idóneo para este sistema.

A continuación, la Tabla 3 muestra las características en la que me he fijado para la elección de este dispositivo.

Procesador	Octa-Core 2.3 GHz
Pantalla	5.8" 146.5mm, full rectangle (5.6" 143.3mm, rounded corners) 2960 x 1440 (Quad HD+)
Memoria	64 GB
RAM	4 GB
Sistema operativo	Android
Compatibilidad VR	Sí
Precio	428€

Tabla 3 Características Samsung Galaxy S8

2.9 Lentes de realidad virtual

Las lentes de realidad seleccionadas han sido unas VRBox por su económico precio y la estética que ofrece.

La Tabla 4 muestra una tabla de las características de las que disponen las lentes.

Compatibilidad de pantallas	3.5" a 6"
Ángulo de visión	85° a 95°
Libertad de movimiento	Total
Precio	11€

Tabla 4 Características de VRBox

3 SISTEMA DE AYUDA AL DIAGNÓSTICO Y TRATAMIENTO DE LA OBESIDAD

A lo largo de este capítulo, se va a detallar todo el procedimiento seguido para obtener el sistema de visualización 3D para el especialista. Este sistema, consiste en una aplicación de escritorio que tendrá el especialista instalado en el ordenador de la clínica, dónde podrá llevar a cabo tareas tales como la visualización 3D del cuerpo de un paciente en distintas sesiones separadas en un espacio temporal, comparativas entre sesiones, consultar los datos de cada sesión etc.

Los modelos 3D utilizados a lo largo del documento han sido obtenidos con el software “ItSeez3D” (ItSeez3D, n.d.). Esto se debe a que el sistema de visualización desarrollado en este trabajo y el sistema de adquisición de los modelos 3D se han realizado en paralelo por lo que no había modelos 3D disponibles para su visualización.

Para desarrollar este sistema con éxito se ha seguido un protocolo formado por diversas fases. Primeramente, se realizó una reunión con el departamento de nutrición y psicología para captar los requisitos que deseaban que tuviera el sistema. Una vez capturado los requisitos se realizó una fase de diseño en la que se elaboró la interfaz gráfica del sistema. Posteriormente, se desarrolló la lógica de la aplicación para darle funcionalidad al sistema. Se realizó una cuarta fase para el desarrollo de una base de datos donde poder almacenar la información de cada paciente. Por último, se estableció un protocolo de comunicación entre el sistema y el servidor encargado de la fase de adquisición con el fin de permitir al especialista realizar desde el sistema de visualización el escaneo de un paciente (Figura 22).

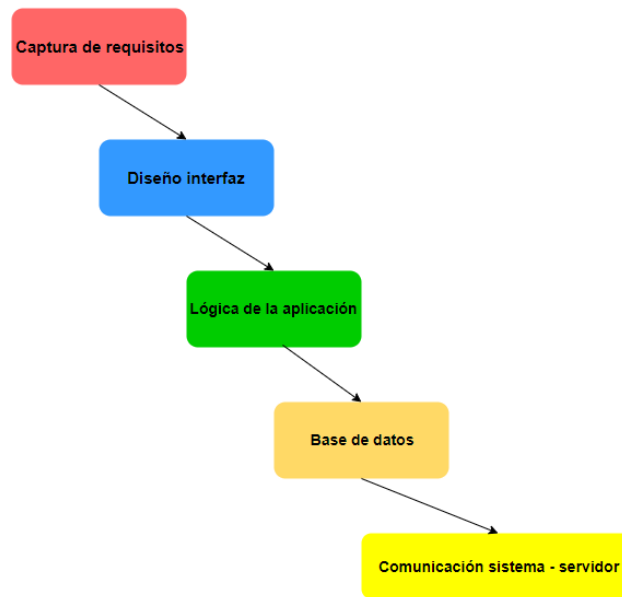


Figura 22 Diagrama del protocolo seguido

3.1 Requisitos

En el libro “Ingeniería del software” (Sommerville, 2005) se realiza una definición sobre un requisito software. “Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste. En el otro extremo, es una definición detallada y formal de una función del sistema.”

Para la captura de los requisitos nos reunimos con el cliente que, en este caso, es el departamento de nutrición y psicología para capturar todos los requisitos que debía tener el sistema. Estos requisitos se han dividido en función de si son requisitos funcionales o no funcionales.

3.1.1 Requisitos funcionales

“Los requerimientos funcionales hacen referencia a la descripción de las actividades y servicios que un sistema debe proveer. Normalmente este tipo de requerimientos están vinculados con las entradas, las salidas de los procesos y los datos a almacenar en el sistema” (Sommerville, 2005). Los requisitos funcionales capturados para este sistema son los siguientes:

- El especialista debe poder visualizar los modelos 3D de los pacientes.
- Al visualizar el modelo 3D del paciente, es necesario que el fondo sea plano para focalizar la atención tanto del especialista como del paciente en su propio cuerpo, evitando elementos que puedan distraer.
- El especialista debe poder visualizar los datos del paciente mientras observa el modelo 3D con el fin de tener toda la información disponible en una misma pantalla.
- El sistema debe permitir girar los modelos 3D para poder visualizarlo desde distintos ángulos.
- El sistema debe permitir hacer zoom sobre cualquier parte del modelo 3D.
- El sistema debe permitir realizar comparativas entre cuerpos 3D de distintas sesiones de un mismo paciente.
- El sistema debe permitir al especialista realizar medidas sobre el cuerpo digital.
- El especialista debe poder dar de alta a nuevos pacientes.
- El especialista debe poder consultar toda la información de un paciente por cada sesión realizada.

3.1.2 Requisitos no funcionales

“Los requerimientos no funcionales describen otras prestaciones, características y limitaciones que debe tener el sistema para alcanzar el éxito. Los requerimientos no funcionales engloban características como rendimiento, facilidad de uso, presupuestos, tiempo de entrega, documentación, seguridad y auditorías internas” (Sommerville, 2005). Los requisitos no funcionales capturado para el sistema son los siguientes:

- La aplicación debe resultar fácil de usar al especialista, ya que es el sistema que va a usar de manera diaria y no tiene alto conocimientos en informática.

- El sistema debe estar comunicado con el servidor que se encarga de escanear los cuerpos 3D de los pacientes.
- El sistema debe ir fluido, sin tirones ni excesivos tiempos de espera.
- El sistema debe estar sincronizado con el sistema de realidad virtual en tiempo real.

3.2 Diseño de interfaz gráfica

El diseño de la interfaz gráfica en una aplicación es una de las partes más importantes en el desarrollo de esta. Esta fase, ha sido un punto de unión con el departamento de nutrición y psicología ya que necesitaban que el sistema de visualización cumpliera diversos requisitos para conseguir el resultado deseado tanto en términos de facilidad para el especialista como para mejorar el tratamiento en los pacientes de obesidad.

Para realizar esta fase con éxito, se han seguido un conjunto de 3 etapas diferenciadas (Figura 23). Primeramente, realicé los mockups de la aplicación y se concertó una reunión con el departamento de nutrición y psicología en la que revisaron los mockups, realizamos algunas modificaciones y obtuvimos los mockups definitivos para el sistema. Tras ello, realicé un diagrama de pantallas apoyado por los mockups elaborados. Por último, desarrollé la interfaz gráfica con Unity y nos volvimos a reunir confirmando que se cumplían todas las expectativas del departamento de nutrición y psicología.

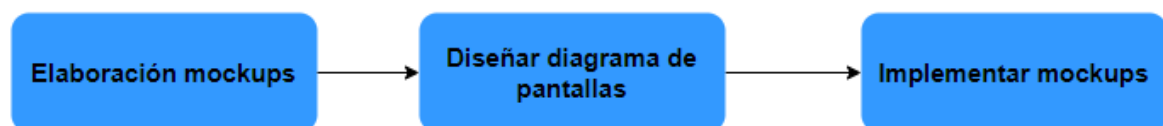


Figura 23 Proceso para elaborar el diseño gráfico

3.2.1 Mockups

El primer paso que hice para comenzar a darle forma a la interfaz gráfica del sistema fue desarrollar los primeros mockups. Para ello, me ayudé de la herramienta Balsamiq.

El primer mockup que elaboré fue el del menú principal (Ver Anexo I, Figura 64). La intención de este menú es que fuera sencillo con dos opciones bien diferenciadas. Por un lado, la posibilidad de cargar un paciente ya registrado o escanear un nuevo cuerpo.

El siguiente mockup que elaboré fue la búsqueda de un paciente (Ver Anexo I, Figura 65). Siguiendo el requisito de facilidad, está formado por un *TextInput* donde rellenar el nombre y apellidos de paciente y un botón de búsqueda.

Uno de los mockups más importantes del sistema era la escena dónde se visualiza el cuerpo del paciente (Ver Anexo I, Figura 66). La parte derecha de la pantalla incluye un panel con los datos del paciente cumpliendo así uno de los requisitos capturados. Además, se ha generado un fondo plano permitiendo así focalizar la atención en el cuerpo.

Por último, elaboré los mockups dedicados a la realización de un nuevo escaneo. Este escaneo se conforma de múltiples pantallas siguiendo el criterio de sencillez. Primero, aparece una pantalla con un formulario a rellenar por el especialista (Ver Anexo I, Figura 67). Tras ella, se pide colocar al paciente sobre la cabina en una postura concreta (Ver Anexo I, Figura 68). Por último, se realiza el escaneo indicando una barra de progreso (Ver Anexo I, Figura 69) y, al acabar el progreso un mensaje de finalización del escaneo (Ver Anexo I, Figura 70).

Tras la elaboración de todos estos mockups, se concertó una segunda reunión para mostrar al departamento de nutrición y psicología los mismos y recibir feedback que ayudó a terminar de limar los mockups.

Hay que destacar que estos mockups han sido la base que ha permitido el desarrollo del sistema, pero existen muchas otras funcionalidades que, con el tiempo, han ido

apareciendo y que, gracias a la metodología ágil aplicada, ha sido bastante sencillo ir mejorando el sistema añadiendo todas las nuevas funcionalidades que se detallarán en mayor profundidad a lo largo del documento.

3.2.2 Diagramas de pantalla

Previo a iniciar el desarrollo de la interfaz gráfica definitiva, realicé un diagrama de pantallas (Figura 24) con el fin de obtener el flujo existente entre las distintas pantallas del sistema, facilitándome así el posterior desarrollo de este.

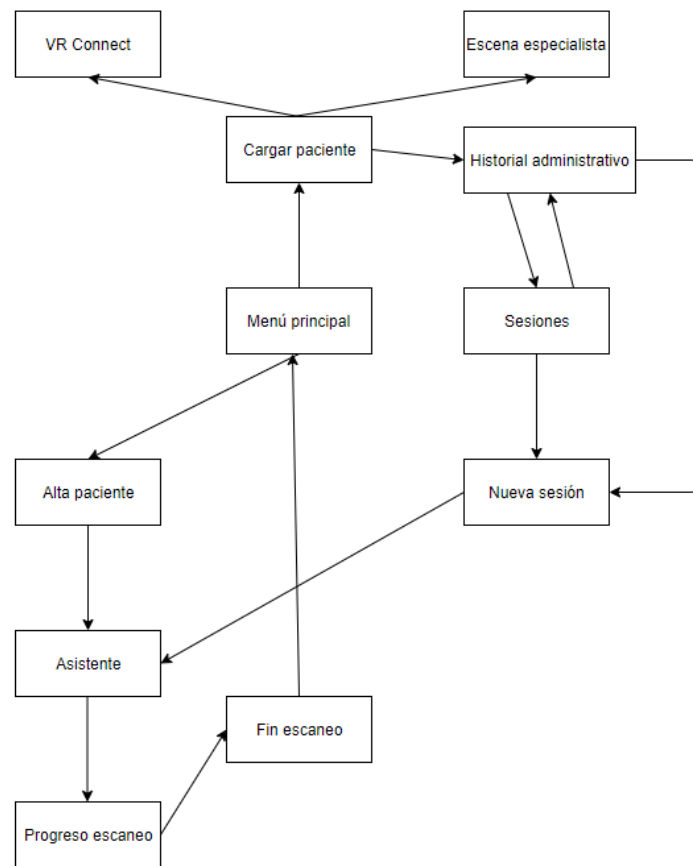


Figura 24 Diagrama de pantallas del sistema de visualización

3.2.3 Implementación de la interfaz

Con los requisitos capturados, los mockups y el diagrama de pantallas elaborados comencé a desarrollar la interfaz gráfica con Unity como herramienta.

La primera pantalla que elaboré fue el menú principal (Ver Anexo II, Figura 71). A diferencia del mockup inicial, ahora existen 4 opciones y la opción “Salir” del mockup se ha transformado por el típico icono de cierre de una aplicación en la esquina superior derecha. Una de las opciones que aparece es la llamada “Visualización paciente” cuyo fin, explicado con profundidad en el bloque “Sistema de realidad virtual”, es sincronizar las gafas de realidad virtual con esta aplicación. Además, se han incluido las opciones de dar de alta al paciente y del acceso a un historial de un paciente, dado que tras distintas reuniones se vio la necesidad de incluirlas, pese a no estar en los primeros mockups.

La pantalla para la búsqueda de un paciente (Ver Anexo II, Figura 72) ha quedado similar a la planteada en el mockup exceptuando que ahora se permite la búsqueda por diversos campos como son el DNI, nombre y apellidos o email.

Para elaborar la pantalla donde el especialista puede visualizar los cuerpos 3D del paciente (Ver Anexo II, Figura 73) ha sido necesario plantear la pantalla en un espacio de 3 dimensiones, a diferencia del resto de pantallas que son de 2 dimensiones. Lo primero que se ha elaborado ha sido el fondo blanco. Para simular ese fondo 3D blanco se han generado un plano vertical y otro horizontal (Figura 25). Ambos planos son más grandes que el field of view de la cámara, de manera que abarca toda la escena. Además, con el fin de dar más realismo a la escena, se ha añadido un foco de luz direccional a la escena generando así las sombras correspondientes a los modelos 3D sobre el plano.

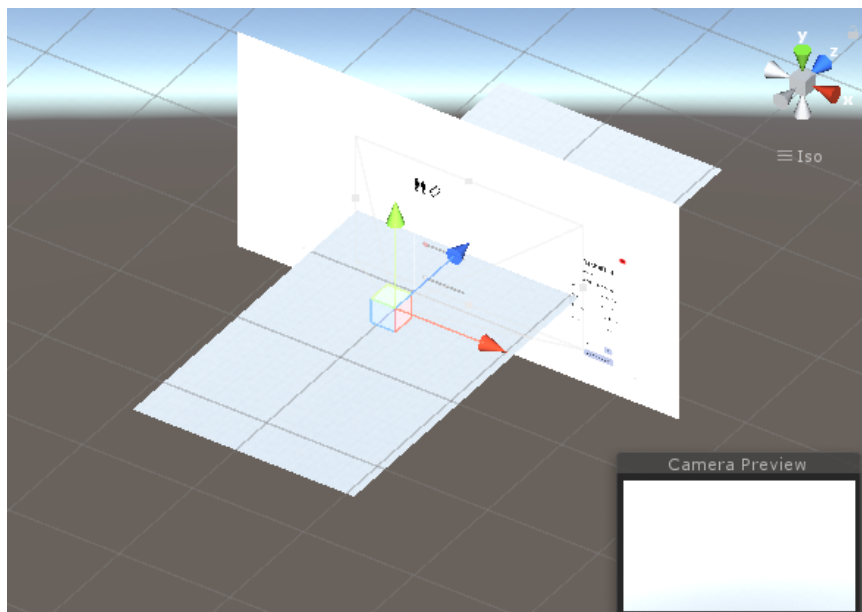


Figura 25 Escena vista desde el exterior para abarcar el plano visto desde la cámara

Para la pantalla del alta de paciente (Ver Anexo II, Figura 74) se ha creado un formulario para que el especialista pueda cumplimentar y, a continuación, pueda realizar el proceso para escanear un cuerpo (Ver Anexo II, Figura 75, Figura 78, Figura 79, Figura 80).

Por último, las pantallas desarrolladas para ver el historial del paciente (Ver Anexo II, Figura 76, Figura 77) comparten la misma estructura y se puede visualizar los datos administrativos del paciente, la información relacionada con cualquier sesión que hayan registradas y, por último, realizar el proceso para escanear el cuerpo.

3.2.4 Problemas encontrados

A lo largo de todo el proceso de diseño de la interfaz gráfica he encontrado algunos problemas. Al desarrollar los mockups, faltaron pantallas por incluir. Para solucionar este problema, conforme fui implementando un conjunto de pantallas, se concertaba una nueva reunión con el departamento de nutrición y psicología para revisar los resultados y recibir feedback para desarrollar nuevas pantallas.

Por otro lado, otro de los problemas que tuve en esta fase fue mi desconocimiento sobre el funcionamiento de Unity, ya que esta etapa fue la primera en la que utilicé esta herramienta. Para solucionarlo, revisé en varias ocasiones el manual de Unity (Unity, 2018) dónde pude formarme paso a paso para elaborar la interfaz gráfica, además de buscar en el foro oficial soluciones a problemas parecidos a los míos.

Por último, entrando en un problema más específico, al desarrollar la pantalla de progreso (Ver Anexo II, Figura 79) me di cuenta de que Unity no dispone de barras de progreso en los elementos UI que lleva por defecto. La solución fue utilizar un slider con 100 valores posibles entre 0 y 1 asignando por software el valor porcentual que deseara en cada momento.

3.3 Lógica de la aplicación

En esta fase, partiendo de la interfaz gráfica desarrollada sin funcionalidades, se han desarrollado todos los comportamientos lógicos que debía tener el sistema de visualización. Para ello, se ha utilizado el sistema de scripting que tiene Unity.

En esta fase, se han seguido 2 etapas (Figura 26). Por un lado, se ha elaborado un diagrama de clases para el sistema de visualización. Una vez diseñado el diagrama de clases del sistema, se han implementado los distintos comportamientos que debe tener siguiendo el diagrama de clases y los requisitos capturados en fases anteriores.

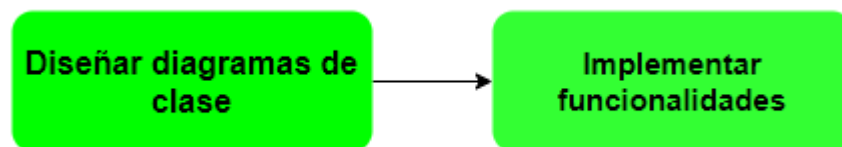


Figura 26 Proceso para elaborar la lógica de la aplicación

3.3.1 Diseño de diagrama de clase

Para elaborar el diseño del diagrama de clase (Figura 27) ha sido necesario tener en cuenta el sistema que utiliza Unity basado en componentes. Los componentes pueden ser propios del sistema de Unity o personalizados por el desarrollador mediante scripts. Gracias al sistema basado en componentes que utiliza Unity, ayuda al desarrollador a cumplir el principio de responsabilidad única. Dicho principio indica que cada clase debe tener responsabilidad sobre una sola parte de la funcionalidad proporcionada por el software y esta responsabilidad debe estar encapsulada en su totalidad por la clase (Winter, 2014).

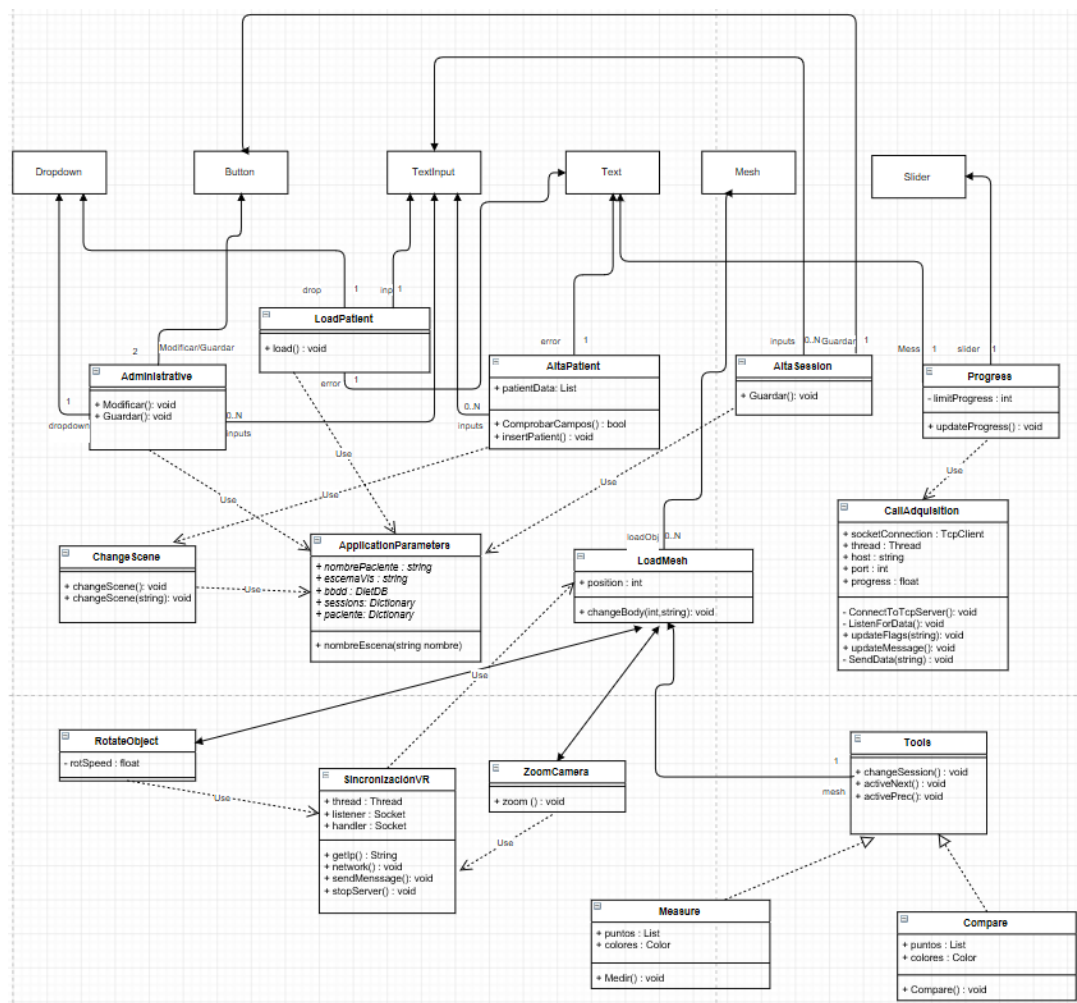


Figura 27 Diagrama de clases del sistema

3.3.2 Implementación de la lógica

La implementación de la lógica del sistema ha sido una de las tareas más largas ya que ha estado en constante actualización. En este apartado, se va a detallar cómo se han desarrollado las distintas funcionalidades que conforman el sistema.

3.3.2.1 Zoom

Para implementar el requisito del zoom se han realizado dos aproximaciones. La primera de ellas consistía en aumentar o disminuir el field of view (fov por sus siglas) vertical de la cámara.

El fov permite variar la extensión del mundo que se puede ver en la pantalla. El fov indica el ángulo de visión que tiene la cámara (Figura 28). Si se reduce el fov, se reduce la extensión del mundo que se puede observar y, al mantener la misma resolución en la pantalla, se genera un efecto de zoom.



Figura 28 Esquema del fov

Esta solución dio los resultados esperados (Figura 29) sin perder nada de calidad del cuerpo que se visualiza. Esta solución se mantuvo hasta la implementación del sistema de realidad virtual dónde, cómo en el apartado de realidad virtual se explicará en detalle, el fov de la cámara del sistema de realidad virtual no se puede modificar, lo que supuso tener que desechar esta solución debido a la necesidad de sincronizar ambos sistemas.

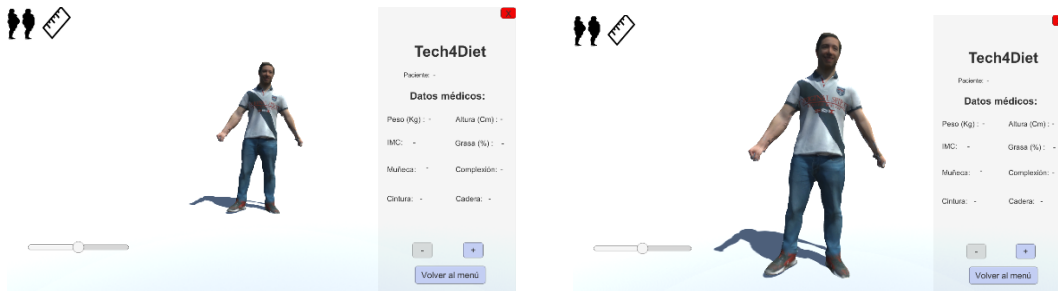


Figura 29 Resultados del fov. Izquierda fov por defecto y derecha fov reducido

Para resolver el problema del zoom y que, fuese compatible con el sistema de realidad virtual, se utilizó una solución que consiste en mover el cuerpo en el eje Z en dirección a la cámara acercándolo o alejándolo en función del deseo del usuario. Esta solución generó unos resultados muy similares a los conseguidos modificando el fov.

3.3.2.2 Rotación del cuerpo

Para la rotación del cuerpo, Unity dispone de un método Rotate() que permite rotar un cuerpo en cualquiera de los ejes X, Y, Z. El comportamiento que debe hacer el cuerpo es girar en el eje Y para que el especialista pueda observar el cuerpo en distintos ángulos. Este giro lo indica el especialista manteniendo el clic derecho del ratón y moviéndolo en el eje X.

Además de tener en cuenta el movimiento del ratón que realiza el especialista, es necesario añadir una variable de velocidad de rotación ya que, de no añadirlo, el especialista tendría que recorrer toda la pantalla de un lateral al otro para girar 360 grados el cuerpo, lo que supondría una difícil manejabilidad. La velocidad de rotación se multiplica al movimiento realizado por el especialista con el ratón para “simular” que ha realizado un movimiento mayor. El valor de la velocidad de rotación se ha

establecido mediante experimentación, observando cual permitía una mejor manejabilidad en el sistema.

Por último, para establecer una rotación independiente a las capacidades computacionales del ordenador dónde se ejecute el sistema, se utiliza la variable delta time. Esta variable indica el tiempo transcurrido entre dos frames y es necesario su uso porque en este tipo de proyectos no existen unos frames fijos por segundo ya que depende del proceso de computación del ordenador. Es por ello, que es necesario utilizar el tiempo que ha pasado desde un frame al siguiente para establecer una rotación proporcional a dicho tiempo consiguiendo así un comportamiento determinista independientemente de las capacidades del ordenador. El resultado obtenido se puede observar en la Figura 30.

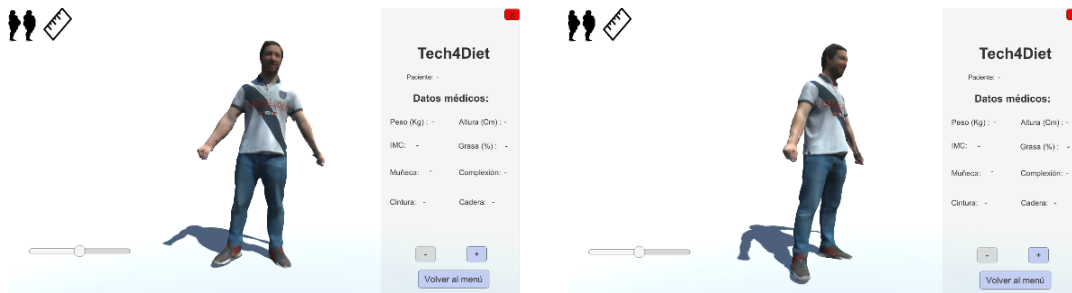


Figura 30 Resultado rotación. A la izquierda sin rotar, a la derecha rotado

3.3.2.3 Factor temporal

Para permitir al especialista variar el factor temporal de las distintas sesiones de un paciente (Figura 31), al cargar la escena de visualización, se renderiza en la escena todos los modelos 3D del paciente y se colocan en la misma posición. Para que únicamente se vea uno de ellos en cada instante, se deja visible uno y el resto se ocultan de manera que cuándo el especialista mueve el slider se oculta el cuerpo visible y se pone visible el cuerpo correspondiente. El motivo por el que se carga al inicio todos los cuerpos pese a que solo se hará visible uno es la manejabilidad del sistema ya que, si solo se carga el que se hace visible en cada momento, genera una sensación de congestión del sistema cada vez que el especialista varía el factor temporal.



Figura 31 Factor temporal. Izquierda cuerpo 1, derecha cuerpo 2

3.3.2.4 Comparativa entre cuerpos

Esta funcionalidad permite al especialista comparar dos cuerpos de sesiones distintas (Figura 32). Para ello, al inicio de la escena se genera un duplicado de cada cuerpo 3D al que se le quita la textura original y se añade una textura de color rojo con una alta transparencia. De esta forma, cuando el especialista realiza una comparativa entre dos de los cuerpos se deja visible un cuerpo con textura y el otro con la textura roja. Al estar en la misma posición en la escena se superponen ambos cuerpos y al disponer de una alta transparencia sólo se puede visualizar las partes que sobresalgan del cuerpo original, permitiendo visualizar las zonas en las que el paciente ha perdido volumen de una sesión a otra.



Figura 32 Comparativa entre dos cuerpos

3.3.2.5 Importación dinámica de los cuerpos en Unity

La importación dinámica de los modelos en Unity ha sido una de las cuestiones más complejas de afrontar del trabajo. Al realizar un proyecto en Unity, lo habitual es que dispongas de los modelos 3D en algún directorio del proyecto antes de ejecutar el sistema, por lo que Unity, de manera automática importa esos modelos 3D en el proyecto. Sin embargo, una particularidad de este trabajo es que no se disponen de los modelos 3D dentro del proyecto, ya que el especialista realiza la adquisición del cuerpo desde el sistema en ejecución por lo que hay que importar dinámicamente los modelos 3D del paciente.

Para solucionar este problema se ha utilizado un plugin creado por el usuario bajo el nick *aaro4130* que permite importar los cuerpos de manera dinámica. Para ello, se envía la ruta donde se encuentra ese modelo y el plugin interpreta el archivo devolviendo dicho cuerpo en memoria para utilizarlo en la escena.

3.4 Base de datos

Las bases de datos hoy en día juegan un papel fundamental en cualquier sistema software que se desarrolla. Es un pilar fundamental ya que permite almacenar conjuntos de datos sistemáticamente para hacer un posterior uso de éste.

Para este sistema, se ha utilizado una base de datos dinámica, es decir, una base de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como creación, borrado y edición de datos, además de las operaciones fundamentales de consulta.

Además de ser dinámica, es una base de datos relacional cuyas bases fueron postuladas en el artículo “*A Relational Model of Data for Large Shared Data Banks*” (Codd, 1970). La decisión de implementar una base de datos relacional se debe a diversos motivos. La primera y más importante se debe a la gran popularidad que tienen este tipo de bases de datos en todo el mundo. Otro de los puntos fuertes de este

tipo de bases de datos es su sencillez de implementación. Además, este modelo evita la redundancia de los datos gracias a las formas normales de sus tablas. El último motivo por el que se decidió implementar este tipo de base de datos es la existencia de un lenguaje de consultas estandarizado como es SQL.

El protocolo seguido para la elaboración de la base de datos estuvo compuesto por diversas etapas (Figura 34). Primero, se realizó una reunión con el departamento de nutrición y psicología para especificar los datos que debía almacenar la base de datos. Tras obtener la especificación, se realizó un diseño de esta. Por último, se implementó una capa de lógica de negocio siguiendo la arquitectura Modelo-Vista-Controlador (MVC por sus siglas) siendo esta capa la perteneciente al Modelo.

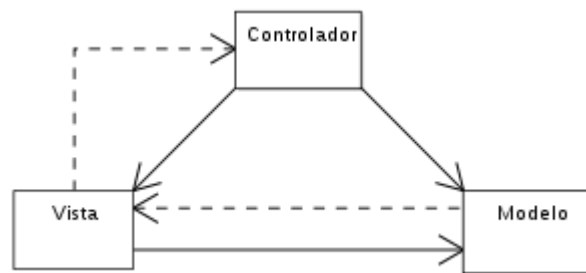


Figura 33 Esquema MVC

Con la implementación de esta capa, el sistema acababa siguiendo un patrón MVC (Figura 33), correspondiéndose la interfaz gráfica, detallada en el capítulo “Diseño interfaz gráfica”, a la capa denominada como vista, la lógica de aplicación, detallada en el capítulo “Lógica de la aplicación” al controlador y esta última capa como el modelo.

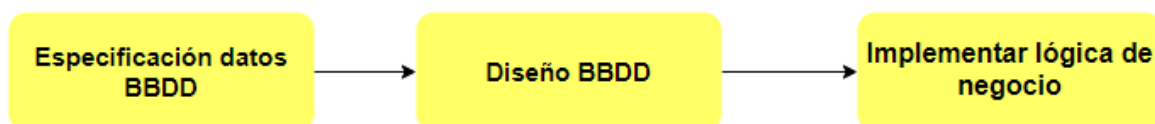


Figura 34 Proceso para elaborar la base de datos

3.4.1 Especificación de datos

Durante la reunión establecida para la especificación de datos, se concretó que para cada paciente necesitaban almacenar dos tipos de datos. Por un lado, los datos administrativos del paciente y por otro, los datos de cada sesión realizada en la clínica.

3.4.1.1 Datos administrativos

Los datos administrativos que establecimos necesarios almacenar por cada paciente en la reunión han sido los comunes en una clínica médica. Estos datos son el nombre, apellidos, edad, dni, sexo, domicilio, población, número de la seguridad social, teléfono y email.

3.4.1.2 Datos por sesión

Los datos por sesión se corresponden a la información respecto a diversas medidas que realiza el especialista con el paciente de obesidad. Por un lado, se utilizan las medidas antropométricas. Estas medidas permiten detectar situaciones de malnutrición de carácter crónico o actual, que afectan fundamentalmente a aspectos energéticos, proteicos y de composición corporal. Las medidas antropométricas se corresponden a los siguientes datos:

- Peso y altura. Estas medidas son primarias por su sencillez y por la información que facilitan en la valoración nutricional. Además, existe una estrecha relación entre el peso corporal y la estatura de una persona.
- IMC. Este índice es un estándar que permite valorar el peso siguiendo la fórmula de la Ecuación 1.

$$IMC = \frac{\text{peso}}{\text{altura}^2}$$

Ecuación 1 Fórmula IMC

Con este índice, el especialista puede categorizar el estado del paciente siguiendo la Tabla 5.

IMC	Categoría
Bajo peso	< 18,5
Peso normal	18,5 – 24,9
Sobrepeso	25,0 – 29,9
Obesidad grado I	30,0 – 34,9
Obesidad grado II	35,0 – 39,9
Obesidad grado III	> 40,0

Tabla 5 Categorías del IMC

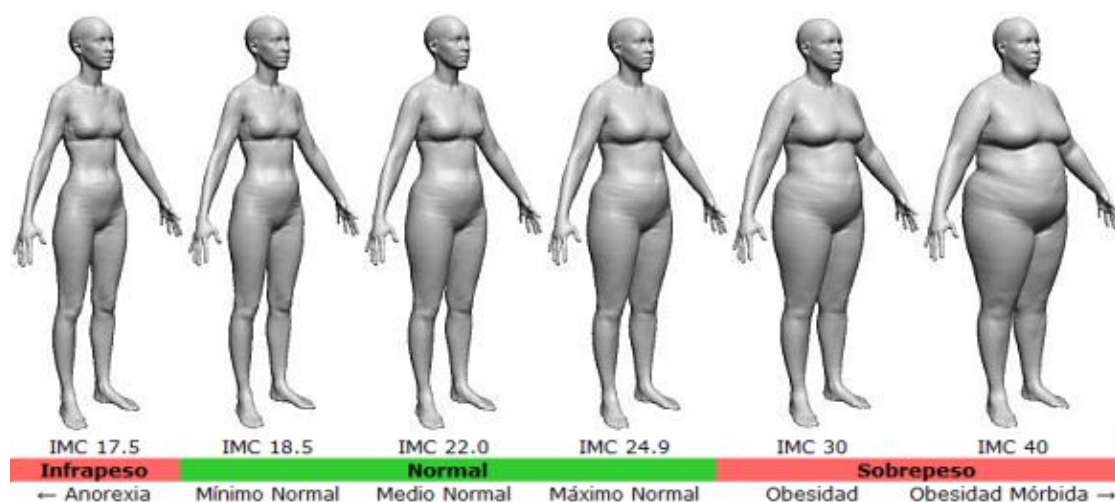


Figura 35 Ejemplo visual del IMC en mujeres

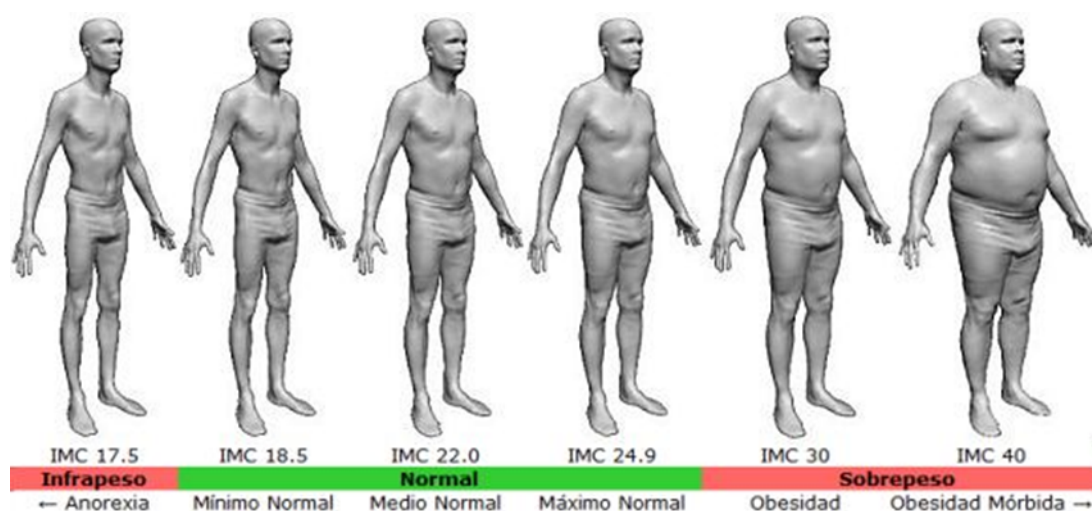


Figura 36 Ejemplo visual del IMC en hombres

- Compleción. Esta medida se puede estimar a partir de la relación entre la estatura y la circunferencia de la muñeca siguiendo la fórmula de la Ecuación 2.

$$Compleción = \frac{talla}{circunferencia\ muñeca}$$

Ecuación 2 Fórmula Compleción

Esta medida se puede categorizar mediante la Tabla 6.

	Varones	Mujeres
Pequeña	$c > 10,4$	$C > 11$
Mediana	$9,6 < c < 10,4$	$10,1 < c < 11$
Grande	$C < 9,6 <$	$C < 10,1$

Tabla 6 Categorías de compleción

- Perímetros. Las medidas más importantes para los especialistas consisten en un conjunto de perímetros.
 - Perímetro del bíceps.
 - Perímetro del antebrazo.
 - Perímetro del muslo.
 - Perímetro gemelar.
 - Perímetro del tórax.
 - Perímetro cintura.
 - Perímetro cadera.
 - Perímetro abdomen.
- ICC. El índice cintura-cadera es un indicador de la distribución de la grasa corporal pone en relación la medida de la cintura con la medida de la cadera para determinar si existe riesgo de contraer determinadas enfermedades asociadas a la obesidad. En la Ecuación 3 se puede observar cómo realizar el cálculo de este índice.

$$ICC = \frac{cintura}{cadera}$$

Ecuación 3 Fórmula ICC

3.4.2 Diseño de la base de datos

Gracias a la especificación anterior, es intuitivo y sencillo el diseño de la base de datos. Se puede apreciar claramente la necesidad de usar dos tablas. Una de ellas almacena los datos administrativos de un paciente y otra los datos de una sesión. La relación entre ellas es que un paciente puede tener múltiples sesiones mientras que una sesión solamente puede pertenecer a un único paciente.

Para la elaboración del diagrama se ha utilizado la notación que utiliza la herramienta draw.io para los diagramas relacionales (Figura 37).

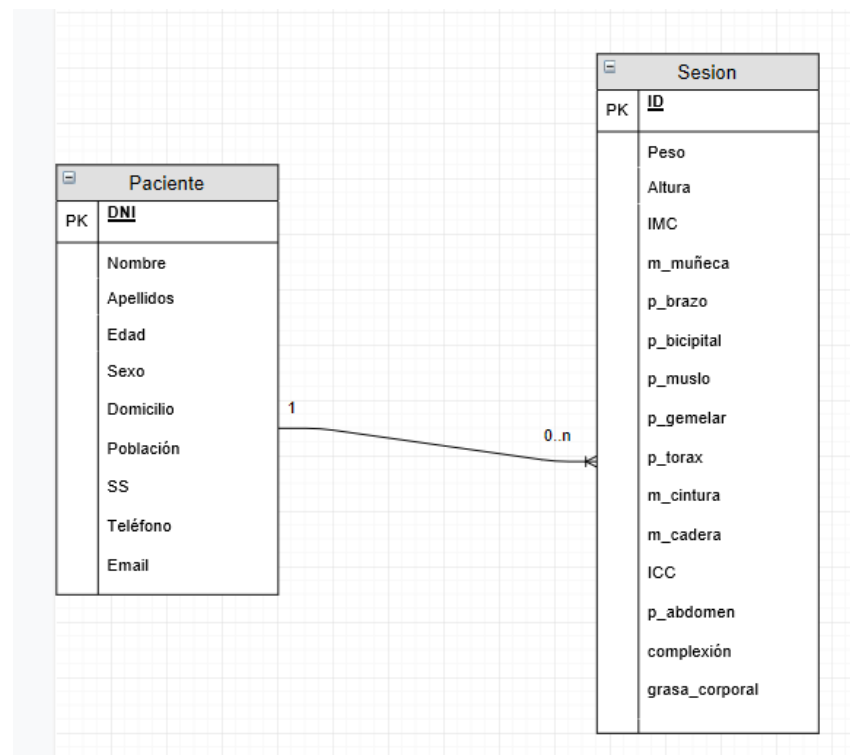


Figura 37 Diagrama de la base de datos

3.4.3 Implementación lógica de negocio

La implementación de la lógica de negocio ha sido una tarea en la que no me he encargado al 100%, ya que he compartido esta tarea junto a mi compañero Jorge Riquelme.

Mi labor en esta etapa ha sido especificar un conjunto de métodos necesarios que me debía proporcionar la lógica de negocio e integrar el desarrollo realizado por mi compañero en el proyecto de Unity.

El conjunto de métodos necesarios se especifica a continuación:

- `BuscarPacientePorCampo(campo)`: Deben existir distintos métodos de búsqueda para los campos DNI, email, y número de la seguridad social del paciente. Si se encuentra el paciente devolverá un diccionario (clase de `c#`) con todos los datos del paciente incluyendo ID y si no devolverá null.
- `BuscarSesionesPaciente(int id)`: Dado un id de un paciente debe devolver un diccionario que tenga como clave el número de sesión y como valor un segundo diccionario con todos los datos de dicha sesión.
- `BuscarDatoSesiones(int id, string campo)`: Dado un id de paciente y el nombre de un campo concreto deberá devolver una lista con todos los datos obtenidos de dicho paciente en ese campo para todas las sesiones.
- Métodos CRUD para ambas tablas.

3.4.4 Problemas encontrados

Uno de los problemas con los que me he topado en esta fase ha sido el integrar el desarrollo realizado por mi compañero Jorge Riquelme en el proyecto porque el desarrollo que hizo lo probó en una base de datos artificial que había creado. Sin

embargo, la base de datos que hay montada tenía algunas variaciones que solventé revisando la implementación de mi compañero y arreglando esas variaciones.

El último problema con el que tuve que lidiar en esta fase fue el bloqueo de las peticiones por parte del firewall de Ubuntu. Inicialmente, cuando hacía alguna petición me rechazaba la conexión e investigando me di cuenta de que el firewall estaba bloqueando la petición. Para solventarlo, añadí en el firewall una excepción para permitir las peticiones a la base de datos.

3.5 Comunicación aplicación – servidor

En esta última fase, el sistema tiene la interfaz gráfica, los comportamientos y la conexión con una base de datos terminada, pero falta establecer una comunicación con el servidor para que el especialista pueda ordenar desde el sistema cuándo escanear a un paciente.

Contextualizando la situación actual, existen dos sistemas. Por un lado, se encuentra el sistema de adquisición dónde un PC tiene conectado 8 cámaras y se encarga de realizar todo el proceso de obtención del cuerpo 3D del paciente y, por otro lado, se encuentra el sistema de visualización desarrollado en este trabajo (Figura 38). El objetivo de esta fase es comunicar ambos sistemas.

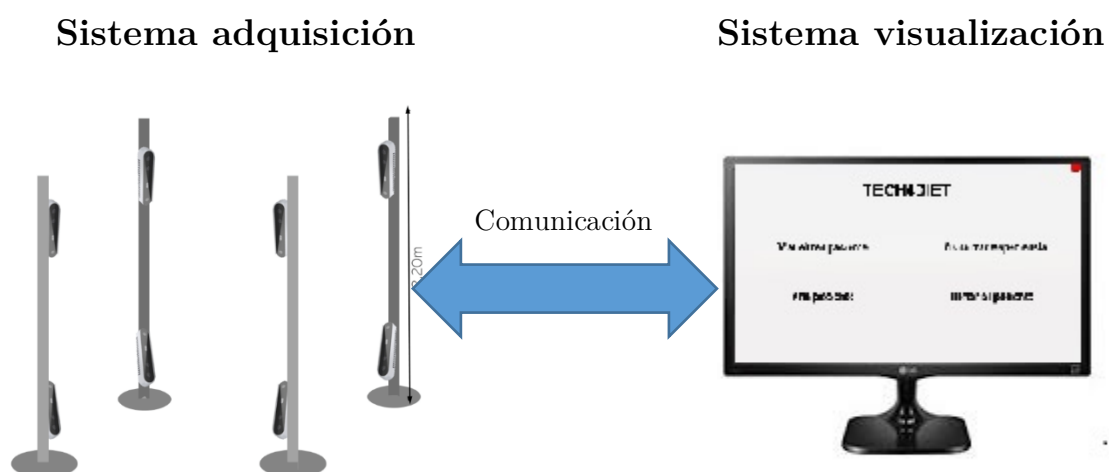


Figura 38 Esquema de los sistemas que hay que comunicar

El protocolo seguido para esta fase se compone de tres etapas (Figura 39). Primero se analizaron los requisitos necesarios para la comunicación del sistema. Después, se diseñó el protocolo de comunicación que debía haber entre el servidor y el sistema de visualización. Por último, se realizó la implementación tanto del servidor como del cliente.

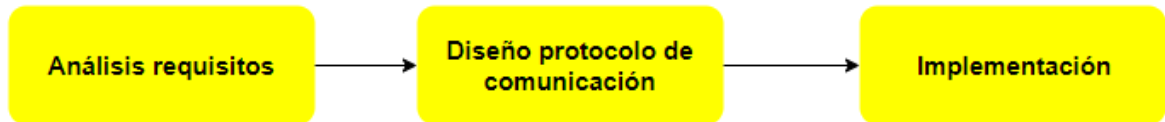


Figura 39 Proceso para elaborar la comunicación aplicación - servidor

3.5.1 Análisis de requisitos

Para realizar este análisis de requisitos nos reunimos y llegamos a la conclusión de que se necesitaba satisfacer dos requisitos.

- Debe ser lo más seguro posible dado que puede haber información médica acerca de los pacientes y es importante preservarla ante cualquier tipo de ataque informático.
- La velocidad de comunicación debe ser bastante rápida para evitar que el paciente esté mucho tiempo manteniendo la misma posición en la cabina.

3.5.2 Diseño protocolo de comunicación

Dado los requisitos previos y el tiempo del que disponía para este trabajo me di cuenta de que la mejor solución posible era que la comunicación se estableciese de manera local ya que ambos sistemas se van a encontrar en la misma clínica. Una red local permite crear un entorno mucho más seguro ya que restringe el acceso al servidor a los dispositivos conectados a la red y, más veloz ya que disminuyen la cantidad de transacciones por la red.

Además, con el fin de evitar descargar por red los modelos 3D que son de gran tamaño, decidí establecer un sistema basado en carpetas compartidas (Figura 40).

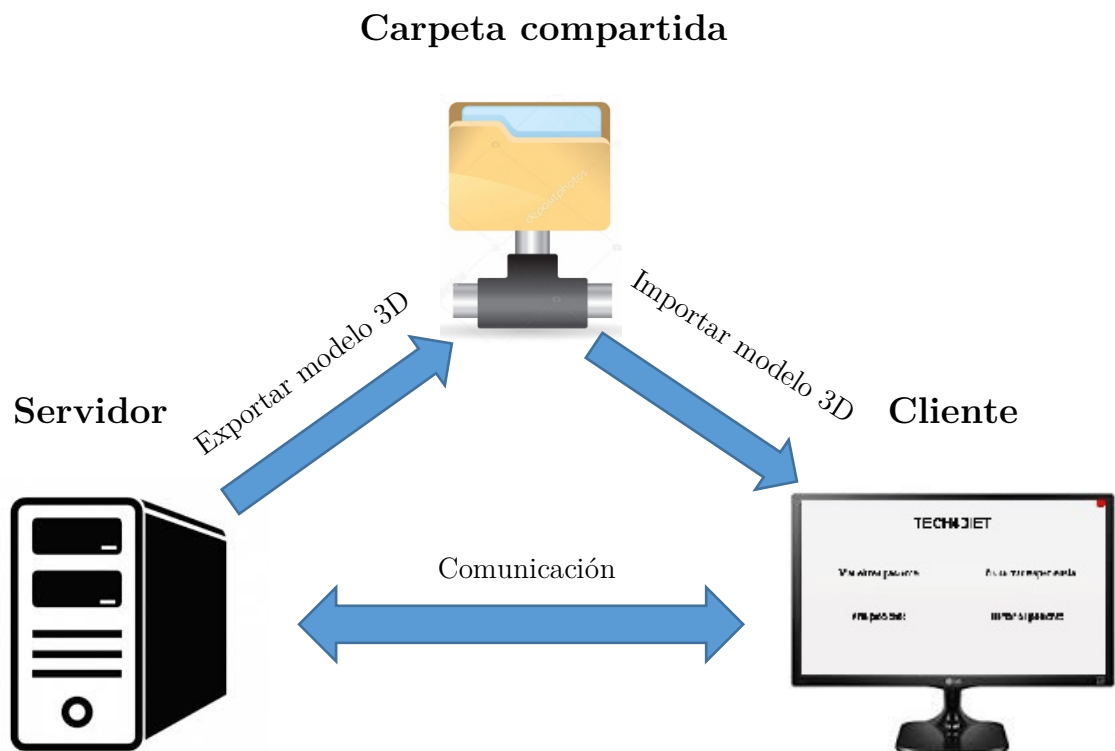


Figura 40 Arquitectura de comunicación

El protocolo de uso (Figura 41) comienza cuando el especialista decide hacer un escáner del paciente mediante el sistema de visualización. El sistema realiza una petición al servidor mediante sockets dónde especifica la ruta dónde desea que se almacene el modelo 3D dentro de la carpeta compartida, el servidor lo procesa y confirma la petición. A partir de aquí, el sistema de adquisición se ejecuta y cuando está listo para realizar la captura le envía una confirmación al sistema de visualización para que se coloque el paciente en la cabina. Una vez colocado el paciente, el especialista lo confirma desde el sistema de visualización que manda dicha confirmación al servidor. El servidor realiza la captura del paciente y lo comunica al sistema de visualización para que el paciente se pueda retirar de la cabina.

Paralelamente, el sistema de adquisición realiza todo el proceso hasta generar el modelo 3D. Durante este proceso, va mandando al sistema de visualización el progreso para reflejarlo en la interfaz. Por último, cuando el sistema de adquisición genera el modelo 3D lo guarda en el directorio especificado dentro de la carpeta compartida y avisa al sistema de visualización de que ha terminado el proceso. A partir de este momento, el sistema de visualización tiene acceso al modelo 3D que se encuentra en la carpeta compartida.

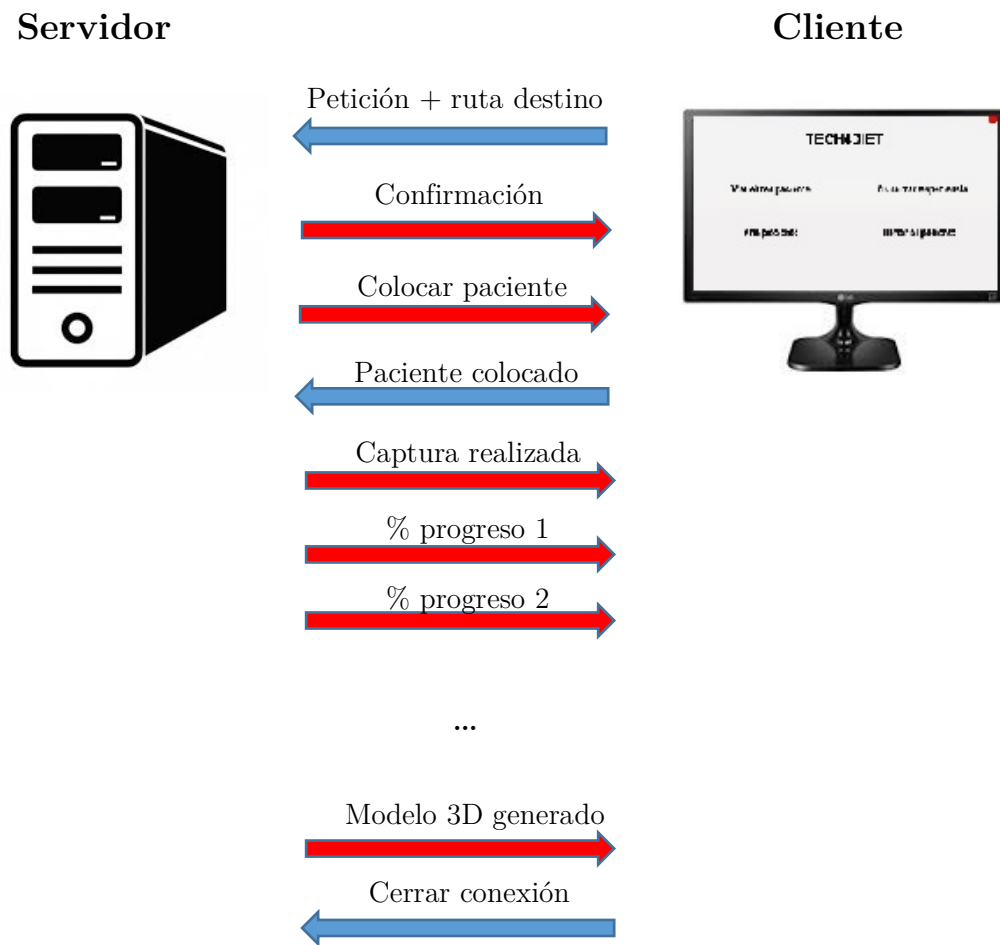


Figura 41 Esquema protocolo de comunicación

3.5.3 Implementación

El desarrollo del protocolo de comunicación se ha establecido mediante sockets. Esta tecnología nos brinda de un mecanismo de bajo nivel que permite establecer un enlace entre dos programas que se ejecutan independientes el uno del otro, normalmente un programa cliente y otro servidor. Por ello, para la implementación del protocolo ha sido necesario desarrollar la parte del servidor en la máquina encargada de la fase de adquisición y la parte del cliente en el sistema de visualización.

3.5.3.1 Servidor

La implementación establecida para el servidor era necesario integrarla con el sistema de adquisición. Este sistema está desarrollado en C++ y posee un paradigma orientado a objetos. El paradigma orientado a objetos es de los más populares. Nació para dar solución a la llamada crisis del software, que permite una alta escalabilidad y que nos proporciona un modelo de abstracción que razona con las técnicas que las personas utilizamos para resolver los problemas (America, 1989).

Para integrar el servidor con la parte de adquisición, se ha encapsulado todo lo relacionado con la parte servidor en una clase, consiguiendo así un bajo acoplamiento y una alta cohesión, principios que se buscan en el correcto diseño orientado a objetos.

Dicha clase, ofrece un método estático que permite mandar mensajes al cliente. Esto ha permitido que, en diversos puntos del programa, se manden porcentajes de progreso.

3.5.3.2 Cliente

La implementación de la parte del cliente se ha integrado en el sistema de visualización en C#. Para ello, ha sido necesario crear un hilo paralelo dónde se ejecutase ya que Unity restringe el uso de cualquier tipo de conexión en el hilo principal porque si se queda colgada la conexión, se quedaría colgado todo el sistema.

4 OBTENCIÓN DE MEDIDAS ANTROPOMÉTRICAS

Uno de los requisitos que debía tener el sistema de visualización era dar la posibilidad al especialista de realizar medidas sobre el modelo 3D del paciente. Este requisito ha sido uno de los mayores retos de este trabajo debido a la complejidad técnica que supone ya que el perímetro de un cuerpo humano es irregular.

Pese a que existen multitud de investigaciones que resuelven la problemática de la obtención de medidas, ninguno de ellos utiliza aproximaciones basadas en las herramientas de un motor gráfico como es Unity.

Para afrontar este desafío, se han realizado hasta tres aproximaciones distintas. La primera de ellas trata de buscar una solución a este problema mediante la búsqueda del vértice más cercano. La segunda aproximación se inspira en el método de Montecarlo para conseguir una estimación precisa de las medidas. Por último, se realiza una tercera aproximación mediante simulación de rayos.

A lo largo de este capítulo se detallarán en profundidad las tres aproximaciones y se hará un estudio comparativo de todas ellas con el fin de ver cuál ofrece mejores resultados.

Cabe destacar que, para la explicación de las tres aproximaciones, se ha utilizado una figura sintética comúnmente conocida como *Armadillo man* (Figura 42).

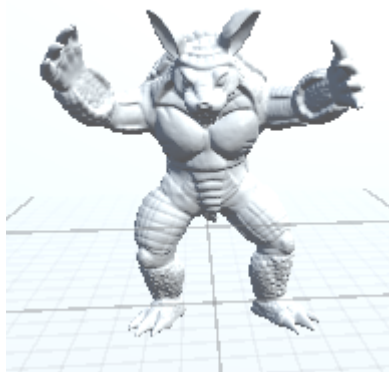


Figura 42 Armadillo man

4.1 Aproximación mediante la búsqueda del vértice más cercano

Esta aproximación surge debido a la estructura que tiene la malla, es por ello por lo que es necesario hacer una pequeña introducción de que es una malla y cuál es su estructura.

Una malla poligonal es un conjunto de vértices, bordes y caras que define la forma de un objeto poliédrico en gráficos por ordenador en 3D y modelado sólido (Tobler & Maierhofer, 2006).

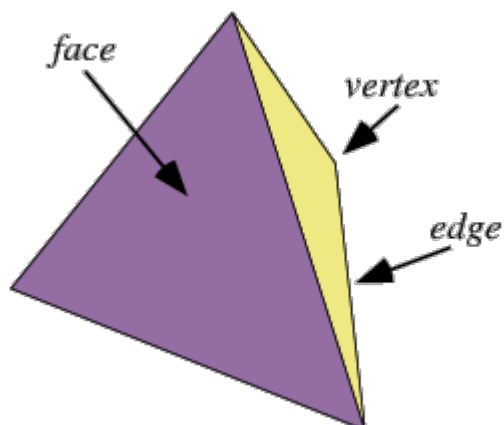


Figura 43 Partes de una malla

Las caras suelen estar formadas por triángulos (malla triangular), cuadriláteros u otros polígonos convexos simples, ya que esto simplifica el renderizado, pero también puede estar compuesto de polígonos cóncavos más generales o polígonos con agujeros. En el caso de este trabajo, mis compañeros de la parte de adquisición han trabajado con el algoritmo de Poisson para la generación de mallas y, por tanto, las caras son triangulares.

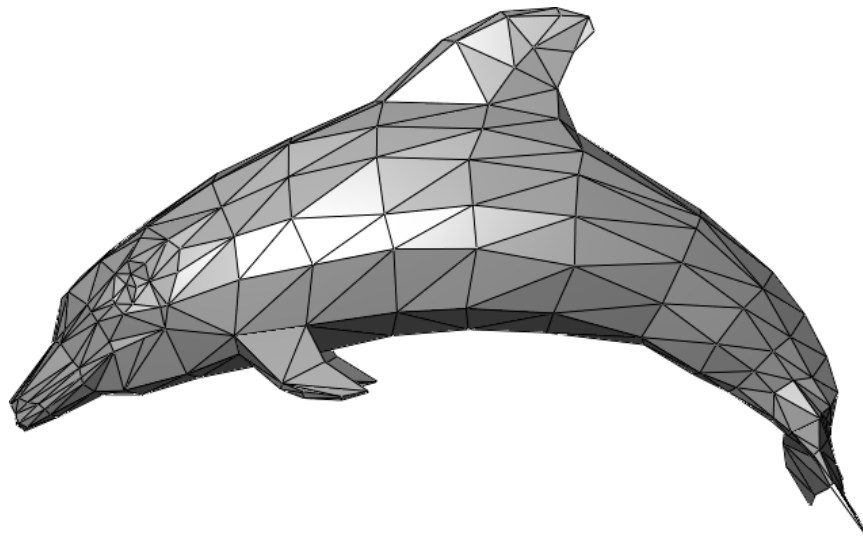


Figura 44 Ejemplo de una malla triangular

El inicio de esta aproximación comienza cuando el especialista marca con el ratón el punto dónde quiere medir el perímetro. Unity permite simular un rayo que viaja desde la cámara hacia la posición que ha marcado el especialista hasta colisionar con el cuerpo. El punto exacto dónde ha colisionado el rayo con la malla se almacena.

Para esta aproximación, nos interesa saber la coordenada Y de dicho punto para conocer a que altura del cuerpo quiere realizar la medida el especialista. Dado que disponemos de información de todos los vértices que forman la malla, seleccionamos todos los vértices cuya coordenada Y se encuentra en un margen determinado de la coordenada del punto registrado por el especialista (Figura 45).

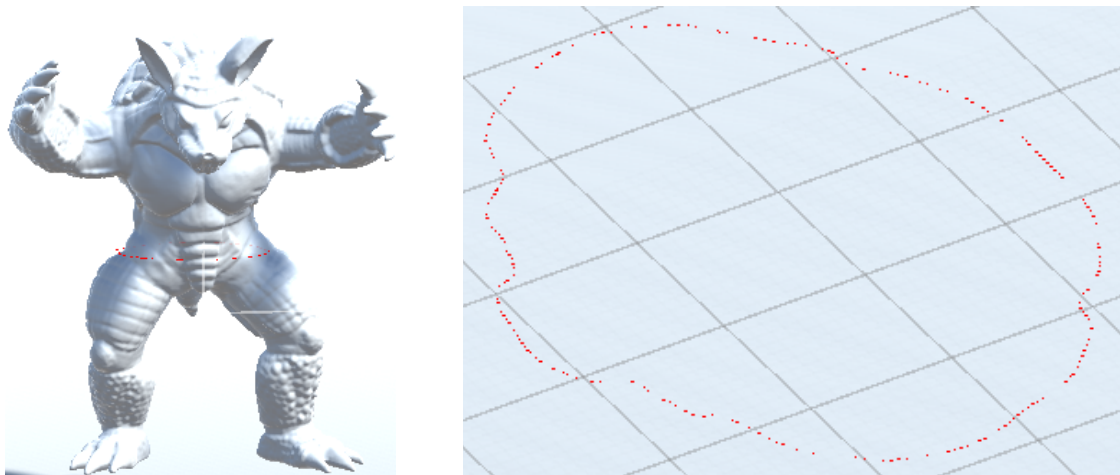


Figura 45 Vértices seleccionados. Izquierda vista frontal, derecha vista sin la malla

Una vez seleccionados los vértices, se dispone de un conjunto de puntos que forman el perímetro del cuerpo, pero desconocemos que puntos se deben unir para formar el perímetro. Esta aproximación, une cada punto con el punto más cercano que existe, midiendo dicha distancia. La suma del conjunto de distancias calculados es el resultado del perímetro para dicho cuerpo.

Profundizando en el algoritmo utilizado para esta aproximación, en dicho algoritmo se realizan los siguientes pasos:

1. Seleccionar todos los vértices de la malla cuya coordenada Y se encuentre en un rango cercano al punto seleccionado por el especialista.
2. Para cada punto seleccionado buscar cual se encuentra a una distancia menor.
3. Si la relación entre los puntos ya fue seleccionada en el orden inverso, es decir, el punto de destino ya seleccionó previamente a este punto como el de distancia más cercana, se selecciona el siguiente más cercano para evitar duplicidades.
4. Acumular la distancia entre los dos puntos seleccionados (Figura 46).

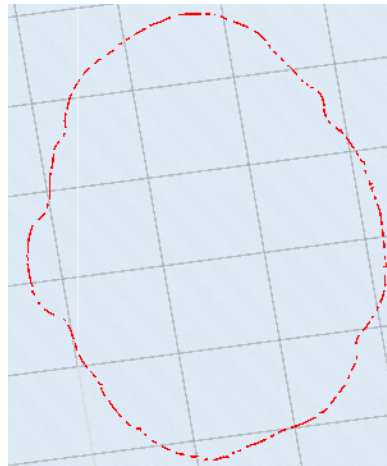


Figura 46 Líneas formadas entre todos los puntos.

4.1.1 Problemas encontrados

Esta aproximación fue la primera que se planteó y que, bajo ciertas circunstancias dio unos resultados bastante precisos. Sin embargo, esta aproximación tiene algunas desventajas. La primera de ellas es que la precisión depende de la cantidad de vértices que haya en esa región del eje Y. Cuántos más vértices se encuentren, mayor es la precisión. Otra de las desventajas que tiene es que si se realizan medidas a una altura donde se encuentran las extremidades no es capaz de asilarlo (Figura 47) y, por tanto, incluye la medida de las extremidades lo que dispara el error de precisión en la medida.

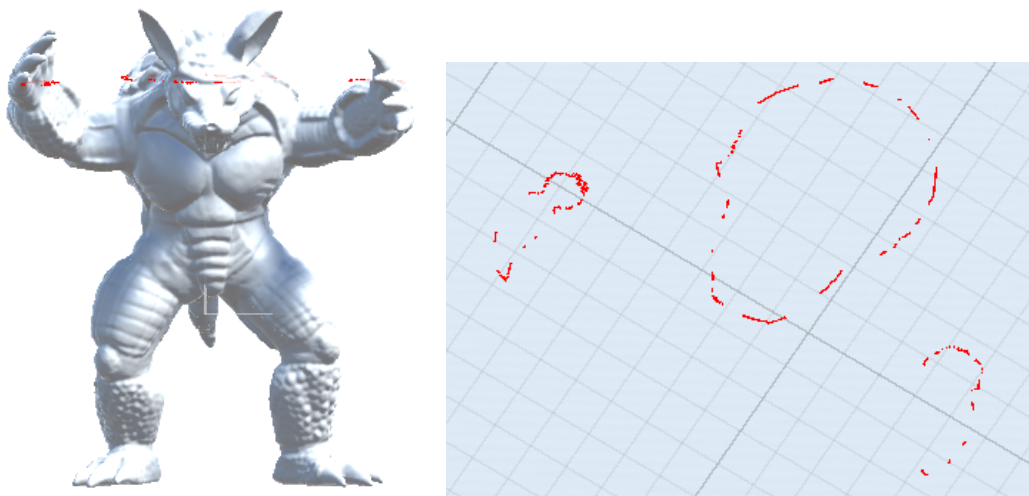


Figura 47 Problema extremidades. Izquierda vista frontal, derecha vista sin la malla

Para solventar este problema, se puede limitar los vértices que se seleccionan en el eje X.

Otro problema es que no siempre el punto más cercano es el adecuado para obtener la distancia correcta (Figura 48). Este problema es el que genera que en el resultado final (Figura 46) existan zonas discontinuas que al final generan imprecisiones en la medida calculada.



Figura 48 Problema de la distancia. Izquierda resultado real, derecha resultado teórico

Por último, otra desventaja es su complejidad computacional, siendo cúbica. Esto supone, que, si hay muchos vértices seleccionados, su coste temporal sea muy alto.

4.2 Aproximación mediante el método de Montecarlo

Esta aproximación consiste en estimar la medida de manera probabilística, mediante el método de Montecarlo (Metropolis, 1987).

Este método es un algoritmo computacional que se basa en muestreos aleatorios repetidos para obtener resultados numéricos. El concepto subyacente es usar la aleatoriedad para resolver problemas que podrían ser deterministas en principio. Tiene multitud de aplicaciones en diversos campos como la ingeniería, las matemáticas, la biología etc.

El inicio de esta aproximación comienza cuando el especialista marca con el ratón el punto dónde quiere medir el perímetro. En ese punto, se genera un plano cuyas dimensiones son conocidas que corta el perímetro de la malla que se desea conocer (Figura 49).

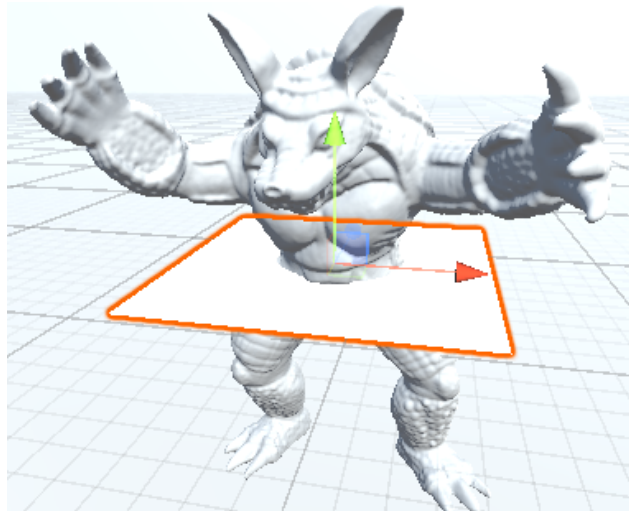


Figura 49 Plano cortando la malla

Una vez generado el plano que corta a la malla, simulamos de manera aleatoria millones de puntos sobre el plano y se observan cuáles han colisionado con el plano y cuáles con el plano y con la malla (Figura 50). De esta manera, se obtiene la probabilidad de que colisione en la malla. Por último, como el área del plano es conocida, el “área” de la malla es el resultado de aplicar el porcentaje de probabilidad al área del plano tal y como se muestra en la Ecuación 4.

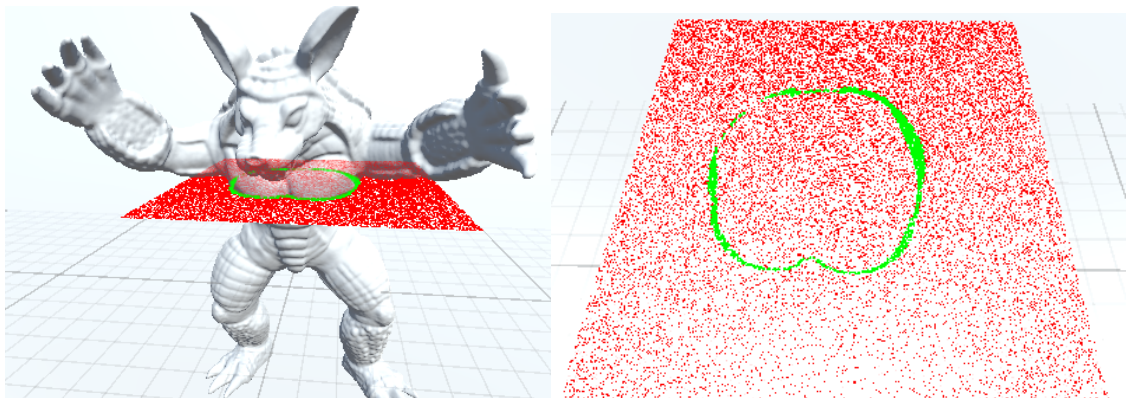


Figura 50 Puntos generados. Izquierda vista frontal, derecha vista sin la malla.

$$\text{área malla} = \frac{\text{área plano} * \text{probabilidad de colisión en la malla}}{100}$$

Ecuación 4 Área de la malla

Siguiendo la teoría, el área obtenida debería ser la que encierra la malla en el punto donde corta el plano. Sin embargo, una malla por dentro está hueca, por lo tanto, los únicos puntos que han colisionado con la malla son los que bordean la malla. Se puede deducir, que en realidad no se ha obtenido el área de la malla, sino el perímetro de la malla que era el objetivo de esta aproximación.

4.2.1 Problemas encontrados

Los principales problemas encontrados se han debido a la integración de este método en Unity. El primer problema, surge con los colisionadores que utiliza Unity. Al realizar la simulación de puntos y comprobar si ha colisionado con la malla o no, me di cuenta de que, Unity para optimizar el funcionamiento de las colisiones, genera una “caja” sobre la malla y esto hace que puntos que no están colisionando los detecte como colisionados (Figura 51).

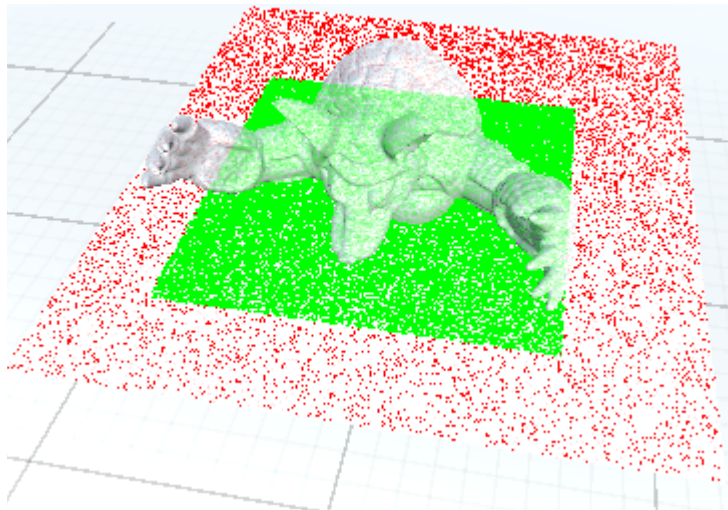


Figura 51 Detección sobre el colisionador de Unity

Para solventar este problema se han simulados rayos de una distancia ínfima que simulen ser puntos, ya que la simulación mediante rayos en Unity si detecta de manera precisa si la colisión se produce con la malla.

El último problema que he detectado ha sido que, dado que el rayo se lanza en trayectoria vertical, en ciertas zonas, debido a la dirección de las normales en ese tramo, no se detectan correctamente las colisiones (Figura 52). Este problema se explica en mayor profundidad en el siguiente apartado ya que debido a este problema se tuvo que desechar una aproximación.

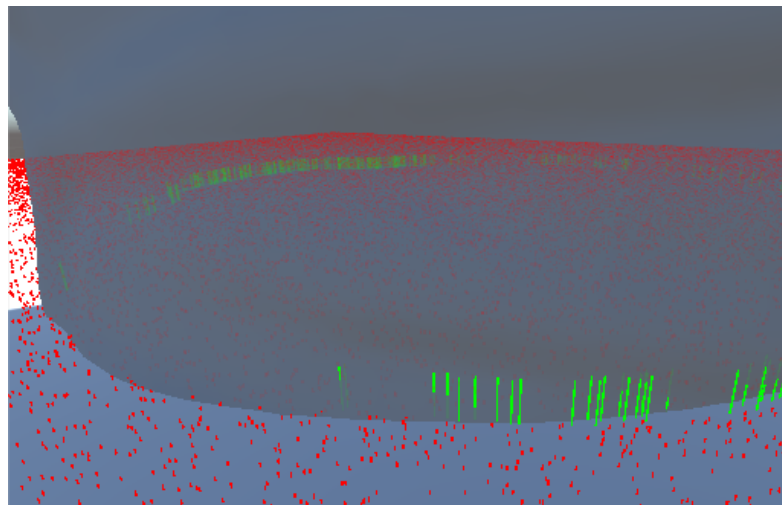


Figura 52 Tramo donde no detecta las colisiones correctamente

4.3 Aproximación mediante simulación de rayos

La última aproximación se basa en la simulación mediante Unity de miles de rayos a lo largo del perímetro que se desea medir. Esta aproximación se ideó de dos maneras distintas. Una de ellos, parte desde el interior de la malla y el otro desde el exterior.

El objetivo en ambos casos es ir obteniendo miles de puntos ordenados que, calculando la recta que les une simule la forma irregular que tenga la malla, consiguiendo obtener una estimación de la medida precisa.

La primera forma en la que se ideó esta aproximación consistía en incorporar un objeto en la posición marcada por el especialista en el interior de la malla. Este objeto simularía miles de lanzamientos de rayos mientras rotaba en el eje Y, de manera que los rayos se lanzasen a lo largo de los 360 grados almacenando los puntos de colisión con la malla.

Esta aproximación no se pudo llegar a implementar por la forma que tiene Unity de renderizar las mallas. Por motivos de optimización, Unity no renderiza las caras interiores de la malla, es decir, las caras cuya dirección de sus normales apunta en el sentido contrario. Esta optimización, provoca que no se detecten los rayos provenientes desde el interior de la malla hacia el exterior.

Esta optimización de Unity se puede comprobar utilizando un plano. Si observamos el plano por delante y por detrás (Figura 53) se aprecia que sólo se renderiza el plano desde uno de los puntos de vista.

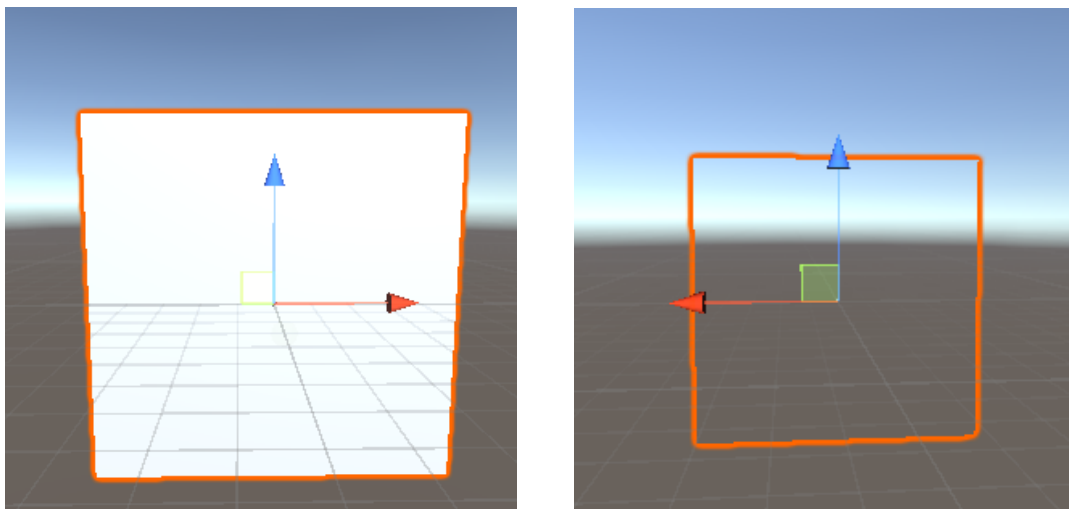


Figura 53 Vista del plano. Izquierda vista frontal, derecha vista frontal trasera

4.3.1 Simulación de rayos desde el exterior

Esta forma consiste en colocar una circunferencia que corte todo el perímetro que el especialista desea medir (Figura 54).

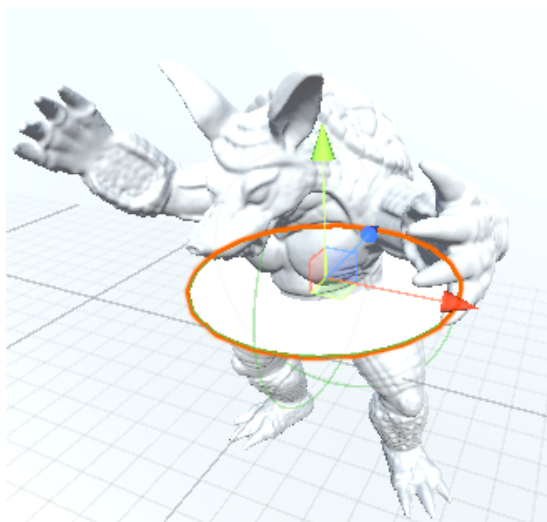


Figura 54 Circunferencia cortando la malla

En la cuerda de la circunferencia se coloca un objeto que simule miles de lanzamientos de rayos mientras realiza una órbita alrededor de la circunferencia, de manera que los rayos se lanzan a lo largo del contorno que se desea medir almacenando los puntos de colisión con la malla (Figura 55).

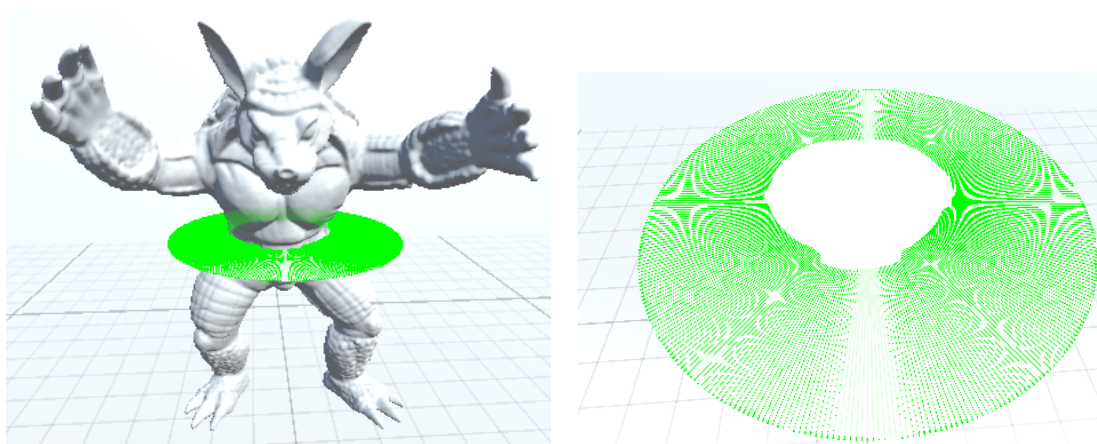


Figura 55 Rayos generados. Izquierda vista frontal, derecha vista sin la malla.

Conforme se van lanzando los rayos, se almacena el punto y se acumula la distancia que hay entre el punto actual y el anterior. La suma de todas las distancias conforma el perímetro deseado.

Esta aproximación ha ofrecido los mejores resultados y tiene grandes ventajas como la mejora de la precisión cuánto más rayos se simulan ya que se obtienen puntos mucho más cercanos. Por otro lado, su coste computacional es lineal respecto al número de rayos lo que permite el cálculo de la distancia de manera casi inmediata.

4.4 Experimentación

Para la realización de la experimentación, se han utilizado cuatro modelos (Figura 56. Estos cuerpos, pueden tener cierto ruido ocasionado por la adquisición que genera un mayor error en la precisión de la medida. Es por ello, que se ha utilizado una cuarta figura sintética cuyas medidas también son conocidas.



Figura 56 Cuerpos usados en la experimentación. Modelos 1, 2, 3 y 4 respectivamente

La Tabla 7 muestra los resultados obtenidos para todos los modelos a la altura del ombligo con cada aproximación, salvo en la esfera que se ha cogido el perímetro central y la medida original que tiene en centímetros.

	Aprox 1	Aprox 2	Aprox 3	Medida real
Modelo 1	102,02	83,52	85,29	87
Modelo 2	120,51	100,67	101,93	103
Modelo 3	105,22	91,08	92,28	94
Modelo 4	78,05	79,12	80,22	80

Tabla 7 Comparativa obtención de medidas

Cómo se puede apreciar, la tercera aproximación es la más precisa obteniendo un error medio de 1,07 cm. Además, el error de todas las aproximaciones con los modelos de cuerpo aumenta debido principalmente a dos motivos. El error generado por los pliegues de la ropa y el ruido proveniente de la malla que conforma el modelo 3D.

4-OBTENCIÓN DE MEDIDAS ANTROPOMÉTRICAS

5 SISTEMA DE REALIDAD VIRTUAL

La tecnología de realidad virtual ha estado orientada desde su origen al sector de los videojuegos y el entretenimiento. Sin embargo, en los últimos años, se está demostrando que esta tecnología puede tener gran utilidad en otros ámbitos como el sanitario. Es por ello, que en este trabajo se ha realizado un sistema de realidad virtual que permite a los pacientes de obesidad ver su progreso de una manera más inmersiva con el fin de conseguir mejorar la adherencia a los tratamientos.

A lo largo de este capítulo, se va a detallar todo el procedimiento seguido para obtener el sistema de realidad virtual. Este sistema, consiste en una aplicación de móvil instalado en la clínica dónde el paciente podrá visualizar su cuerpo en diferentes sesiones mediante realidad virtual.

El proceso seguido para el desarrollo del sistema (Figura 57) fue realizar una reunión con el departamento de nutrición y psicología para capturar algunos requisitos específicos de este sistema y se desarrollaron las funcionalidades requeridas para cumplir los requisitos capturados del sistema.



Figura 57 Proceso para elaborar el sistema de realidad virtual

5.1 Requisitos

De la misma manera que se realizó la captura de requisitos para el sistema de visualización en la página 30. Estos requisitos se han dividido en requisitos funcionales y requisitos no funcionales.

5.1.1 Requisitos funcionales

Los requisitos funcionales capturados para el sistema de realidad virtual son los siguientes:

- El cuerpo debe girar con el movimiento de cabeza del paciente.
- La cámara debe estar fija, es decir, la cámara no debe moverse acorde al movimiento de cabeza.
- Debe sincronizarse en tiempo real con el sistema de visualización del especialista, de manera que el especialista vea en su pantalla lo que ve el paciente con las gafas de realidad virtual.
- El escenario debe ser el mismo utilizado en el sistema de visualización del especialista.
- El sistema de realidad virtual debe permitir que el especialista pueda interactuar desde el sistema de visualización.

5.1.2 Requisitos no funcionales

Los requisitos no funcionales capturados para el sistema de realidad son los siguientes:

- La aplicación debe tener un tamaño lo más pequeño posible.
- La comunicación entre el sistema de visualización y realidad virtual debe efectuarse en local.

5.2 Implementación

El sistema de realidad virtual, al igual que el de visualización, ha sido desarrollado mediante Unity. Este motor, cuenta con una API llamada XR que permite el desarrollo de sistemas de realidad virtual (VR), de realidad aumentada (AR) y de realidad mixta (MR). Además, se ha reutilizado la escena de visualización desarrollada en el sistema

de visualización del especialista añadiendo nuevos scripts para cumplir las nuevas funcionalidades de este sistema.

A lo largo de este bloque, se van a detallar las distintas funcionalidades desarrolladas para cumplir los requisitos capturados para este sistema.

5.2.1 Transmisión del movimiento de cabeza al objeto 3D

Uno de los requisitos capturados fue que, con el movimiento de cabeza del usuario, se girase el cuerpo 3D (Figura 58). Para resolverlo, Unity permite conocer gracias a la API XR la posición de la cabeza en cada instante. Dado que el único eje que se desea que rote es el Y, en cada frame se ajusta la rotación Y del cuerpo 3D a la rotación Y de la cabeza del usuario.

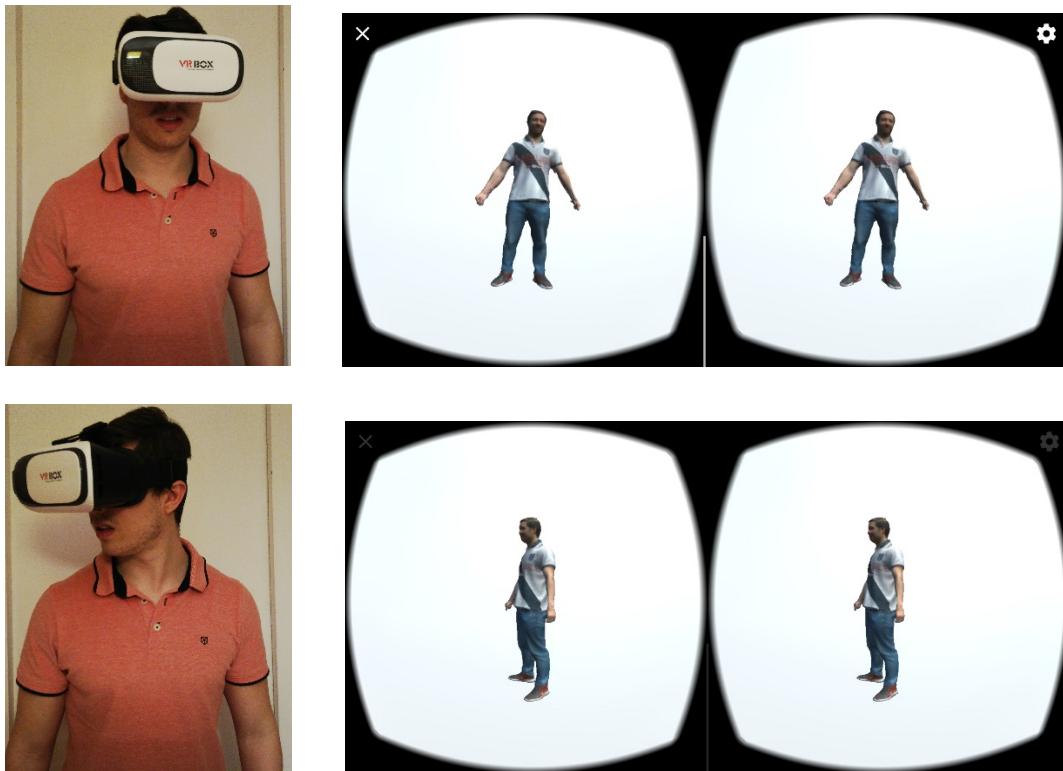


Figura 58 Muestra de la transmisión de movimiento.

5.2.2 Bloqueo de la cámara en realidad virtual

Uno de los grandes problemas a la hora de desarrollar las funcionalidades ha sido el bloqueo de la cámara en realidad virtual.

Tanto Unity, como la mayoría de las herramientas que permiten el desarrollo de realidad virtual, actualmente siguen una política de no permitir al desarrollador realizar ajustes de la cámara, sino que la propia API es la que se encarga de dicho cometido. Esta política se debe a los mareos que puede conllevar un mal ajuste de la cámara en el usuario final.

En el caso particular de Unity, tras investigar cómo funcionaba internamente, cuando realizas una actualización de la cámara en un frame determinado, la API realiza una última actualización actualizando los ajustes de la cámara. Por tanto, aunque no se quiera rotar la cámara, al final es la API el que lo realiza.

Para solventar este problema se ha desarrollado un script que se encarga de aplicar la rotación inversa que ha hecho el paciente (Figura 59). De esta forma, en el mismo frame, se aplica primero una rotación inversa a la cámara y, sobre esa posición, la API actualiza la rotación de la cámara en base a la rotación realizada por el usuario. De esta forma, el script neutraliza la rotación de la API en el mismo frame y, da la sensación de que la cámara está siempre en la misma posición.

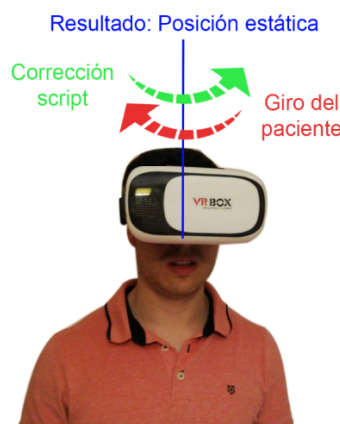


Figura 59 Funcionamiento del bloqueo de cámara

5.2.3 Sincronización con aplicación de escritorio

La última funcionalidad consiste en la comunicación en tiempo real entre el sistema de visualización del especialista y el sistema de realidad virtual para el paciente (Figura 60).

Por motivos de seguridad, el entorno donde se pueden comunicar ambos sistemas es local. El uso de ambos sistemas está orientado a la clínica. Por ello, tanto el dispositivo móvil con la aplicación instalada, como las lentes de realidad virtual y el sistema de visualización instalado en el ordenador del especialista deben estar en la propia consulta.

Sistema visualización

Realidad virtual

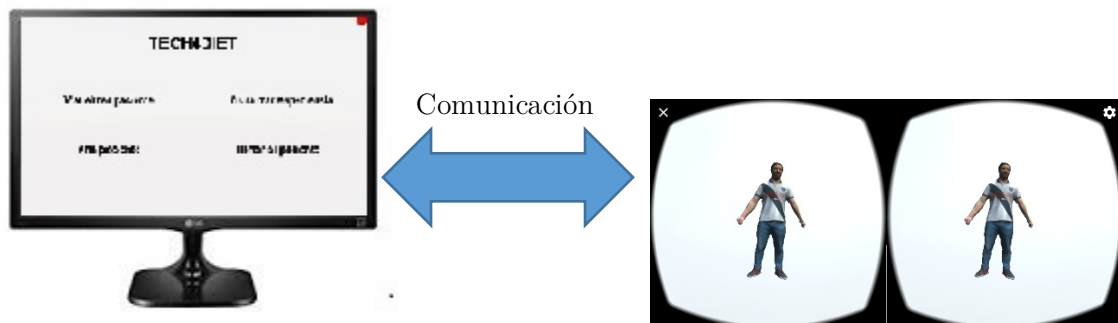


Figura 60 Esquema de los sistemas que hay que comunicar

5.2.3.1 Protocolo de comunicación

La comunicación establecida parte de un modelo cliente-servidor dónde el sistema de visualización hace de servidor y el sistema de realidad virtual de cliente. Desde el sistema de visualización hay una opción para visualizar mediante realidad virtual en la que espera que el sistema de realidad virtual le haga una petición (Figura 61).



Figura 61 Pantalla de espera.

Una vez se abre el sistema de realidad virtual realiza dicho contacto y comienza un protocolo de comunicación (Figura 62). El sistema de visualización envía los modelos 3D que necesita que se deben cargar en el sistema de realidad virtual. Tras enviar todos los modelos, en ambos sistemas se cargan los cuerpos en las mismas posiciones (Figura 63).

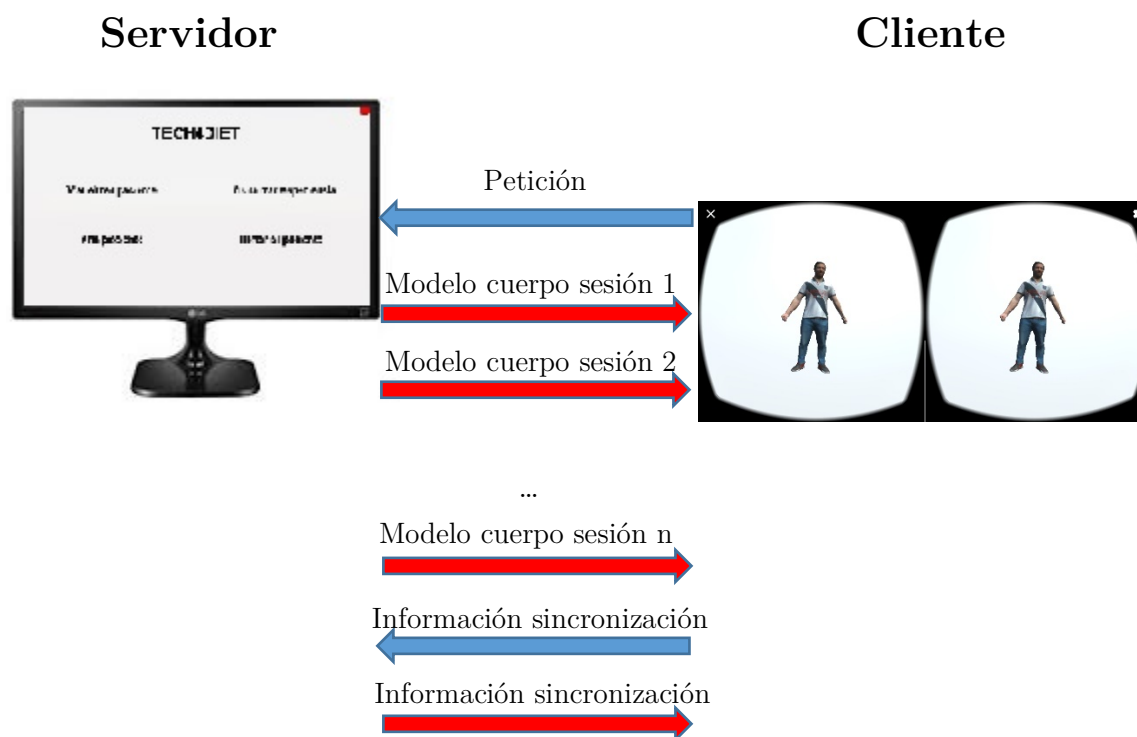


Figura 62 Protocolo de comunicación

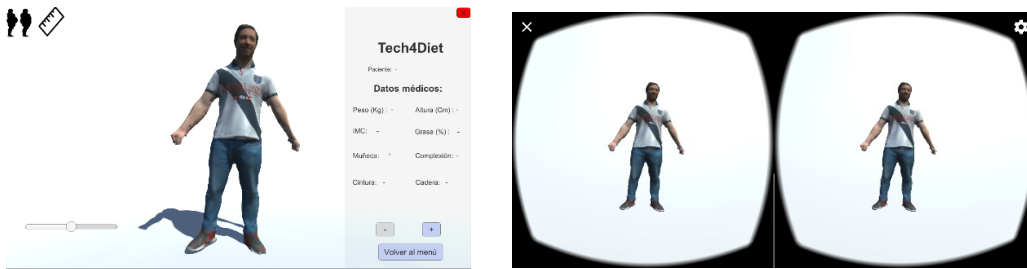


Figura 63 Sincronización. Izquierda aplicación escritorio, derecha aplicación VR.

Por último, existe una comunicación bidireccional en la que ambos sistemas se mandan información para mantener sincronizado en todo momento los dos sistemas.

5.2.3.2 Información para la sincronización

La información transmitida entre ambos sistemas es importante para mantener los sistemas sincronizados en todo momento. Dado que es bidireccional, se va a especificar tanto la información que envía el sistema de realidad virtual, como la información que envía el sistema de visualización.

- Sistema de realidad virtual. La información que manda este sistema consta de la rotación realizada por el paciente con la cabeza. Esto supone que, solo la persona que está usando el sistema de realidad virtual puede rotar el objeto, ya que el sistema de visualización únicamente lee la información y lo actualiza acorde a ella.
- Sistema de visualización. La información que manda este sistema está compuesta por 2 valores. El primero de ellos indica la posición en el eje Z del cuerpo, es decir, el zoom que hay aplicado sobre el cuerpo. El otro valor indica que sesión se debe estar visualizando. Esto supone que, el especialista es el que puede manejar tanto el zoom como la sesión que se debe visualizar en el sistema de realidad virtual.

5.2.3.3 Problemas encontrados

El principal problema encontrado ha sido la cantidad de información transmitida en tiempo real entre ambos sistemas. En un segundo se están mandando de media 80 mensajes con información. El problema que supone es que los sockets bajo esa gran cantidad de información transmitida se acaban saturando. Para solucionar este problema, se le pone un timing para el envío de información, de manera que, pese a que es menos preciso, apenas se nota en el resultado final y mejora la comunicación entre ambos sistemas.

6 CONCLUSIÓN

6.1 Conclusiones

El trabajo desarrollado en este proyecto responde a la necesidad detectada en el estado del arte del uso de tecnologías de visión 3D y realidad virtual para la mejora en la adherencia a tratamientos en pacientes de obesidad. En consecuencia, se ha desarrollado un sistema de visualización 3D y realidad virtual para su uso en clínicas dietéticas que está previsto utilizar en centros de salud en el marco del proyecto de investigación en el que se ha desarrollado el trabajo.

De forma más concreta el trabajo ha abordado una serie de objetivos específicos que fueron planificados en tareas y que han culminado con los siguientes logros.

Primero, se ha desarrollado un sistema de visualización realista de modelos 3D desde cero, realizando todo el proceso necesario para su elaboración. Se han realizado las tareas de captura de requisitos, desarrollo de la interfaz gráfica, desarrollo de la lógica del sistema, implementación de una BBDD y establecer la comunicación entre sistema y servidor. Tras realizar dichas tareas, se concluye que se ha logrado el primer objetivo establecido en el documento.

Posteriormente, se han realizado un conjunto de aproximaciones que permiten obtener medidas precisas sobre el modelo 3D. Este objetivo ha supuesto el mayor reto científico del trabajo y se ofrecen tres aproximaciones para abordar este problema con

Unity como herramienta. Estas aproximaciones han generado buenos resultados por lo que se concluye que se ha logrado el segundo objetivo establecido en el documento.

Por último, se ha desarrollado un sistema de realidad virtual para visualizar de manera realista los modelos 3D. Se han realizado tareas como el análisis de dispositivos y el desarrollo de la aplicación móvil y la sincronización con el sistema de visualización 3D. Tras realizar dichas tareas, se concluye que se ha logrado el tercer objetivo establecido en el documento.

La principal aportación realizada en este trabajo es el desarrollo de todo un sistema que está previsto utilizar en el marco del proyecto de investigación para la futura investigación del departamento de nutrición y psicología de los beneficios que ha aportado dicho sistema en los pacientes de obesidad.

Este trabajo se enmarca en el proyecto de investigación titulado “Modelado y visualización 4D del cuerpo humano para la mejora de la adherencia al tratamiento dietético nutricional de la obesidad” financiado por el ministerio de economía, industria y competitividad con referencia “TIN2017-89069-R”.

6.2 Líneas futuras

Este trabajo es una aportación más para que tanto yo, como cualquier otra persona pueda continuarlo y, sobre todo, mejorarlo. Con el fin de orientar a toda aquella persona que desee continuar este trabajo, se plantean algunas líneas futuras para mejorar este sistema.

- Incorporar medidas del cuerpo 2D, 3D y 4D.
- Desarrollar un sistema mediante técnicas de machine learning, que permita predecir la evolución futura del cuerpo del paciente.
- Desarrollar un sistema mediante técnicas de machine learning, que identifique puntos sobre el cuerpo 3D tales como la cintura, los hombros etc. donde se realicen medidas automáticamente.

- Encriptar toda la información que se transmite en red pudiendo alojar así el servidor de manera remota.
- Experimentar nuevas técnicas que permitan mejorar la precisión de las medidas sobre cuerpos digitales.

6.3 Conclusiones personales

Las conclusiones personales que saco de este proyecto son múltiples y satisfactorias.

Por un lado, he aprendido a desarrollar un proyecto real desde cero asumiendo responsabilidades. Este aprendizaje me será muy útil en mi carrera profesional para abordar nuevos proyectos.

Por otro lado, la oportunidad de trabajar con un equipo altamente cualificado es una de las mejores experiencias que me llevo de este trabajo, con grandes aprendizajes en la materia.

Además, este trabajo ha supuesto mi primer, y espero que no sea el último, contacto con el mundo de la investigación. Lo que me ha permitido adquirir nuevas habilidades tanto de redacción, como de investigación de trabajos similares y de pensamiento crítico.

Por último, me llevo grandes aprendizajes respecto a nuevas tecnologías punteras como la realidad virtual que, hasta este proyecto, no había tenido la oportunidad de trabajar con ellas.

6-CONCLUSIÓN

7 REFERENCIAS

- , P. R. A. (2010). Application of 3D body scanning technology to human measurement for clothing Fit. *International Journal of Digital Content Technology and Its Applications*. <https://doi.org/10.4156/jdcta.vol4.issue7.6>
- Ajie, W. N., & Chapman-Novakofski, K. M. (2014). Impact of computer-mediated, obesity-related nutrition education interventions for adolescents: A systematic review. *Journal of Adolescent Health*. <https://doi.org/10.1016/j.jadohealth.2013.12.019>
- America, P. (1989). Object-oriented software construction. *Science of Computer Programming*. [https://doi.org/10.1016/0167-6423\(89\)90034-8](https://doi.org/10.1016/0167-6423(89)90034-8)
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*. <https://doi.org/10.1145/362384.362685>
- Fernando Llopis Pascual. (2015). Guerreros Ágiles. Gestión de equipos SCRUM eBook: Fernando Llopis: Amazon.es: Tienda Kindle. Retrieved May 25, 2019, from <https://www.amazon.es/Guerreros-Ágiles-Gestión-equipos-SCRUM-ebook/dp/B019H2LIA2>
- Hall, K. D., & Jordan, P. N. (2008). Modeling weight-loss maintenance to help prevent body weight regain. *American Journal of Clinical Nutrition*. <https://doi.org/10.3945/ajcn.2008.26333>
- He, Q., Ji, Y., Zeng, D., & Zhang, Z. (2018). Volumeter: 3D human body parameters measurement with a single Kinect. *IET Computer Vision*. <https://doi.org/10.1049/iet-cvi.2017.0403>
- Hussein, M., & Nätterdal, C. (2015). The Benefits of Virtual Reality in Education: A

- Comparison Study. *University of Gothenburg, Chalmers University of Technology*.
- Intel. (2015). Tecnología Intel® RealSense™. Retrieved May 24, 2019, from <https://www.intel.es/content/www/es/es/architecture-and-technology/realSense-overview.html>
- ItSeez3D. (n.d.). #1 Mobile 3D Scanner App for iPad | itSeez3D. Retrieved May 31, 2019, from <https://itseez3d.com/>
- Kazhdan, M., Bolitho, M., & Hoppe, H. (2006). Poisson surface reconstruction. *ACM Transactions on Graphics*. <https://doi.org/10.1145/2487228.2487237>
- Kuzmar, I., Rizo, M., & Cortés-Castell, E. (2014). Adherence to an overweight and obesity treatment: how to motivate a patient? *PeerJ*. <https://doi.org/10.7717/peerj.495>
- Lin, J. Der, Chiou, W. K., Weng, H. F., Tsai, Y. H., & Liu, T. H. (2002). Comparison of three-dimensional anthropometric body surface scanning to waist-hip ratio and body mass index in correlation with metabolic risk factors. *Journal of Clinical Epidemiology*. [https://doi.org/10.1016/S0895-4356\(02\)00433-X](https://doi.org/10.1016/S0895-4356(02)00433-X)
- Metropolis, N. (1987). The beginning of the Monte Carlo method. *Los Alamos Science Special Issue*.
- Roseman, M. G., Riddell, M. C., & Haynes, J. N. (2011). A Content Analysis of Kindergarten-12th Grade School-based Nutrition Interventions: Taking Advantage of Past Learning. *Journal of Nutrition Education and Behavior*. <https://doi.org/10.1016/j.jneb.2010.07.009>
- Rusu, R. B., & Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). *Proceedings - IEEE International Conference on Robotics and Automation*. <https://doi.org/10.1109/ICRA.2011.5980567>
- Schwaber, K., & Sutherland, J. (2011). Scrum Guide. In *Arbeit*. <https://doi.org/10.1053/j.jrn.2009.08.012>
- SDK Tools / Android Developers*. (n.d.). Retrieved from <https://developer.android.com/tools/sdk/tools-notes.html>
- Sommerville, I. (2005). Ingeniería del software. In *danielr.obolog.es*.
- Tobler, R. F., & Maierhofer, S. (2006). A mesh data structure for rendering and

- subdivision. *Proceedings of WSCG (International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision)*.
- Tomasi, C., & Manduchi, R. (2002). *Bilateral filtering for gray and color images*. <https://doi.org/10.1109/iccv.1998.710815>
- Treleaven, P., & Wells, J. (2007). 3D body scanning and healthcare applications. *Computer*. <https://doi.org/10.1109/MC.2007.225>
- Unity. (2018). Unity User Manual (2018.3) - Unity Manual. Retrieved May 25, 2019, from <https://docs.unity3d.com/es/current/Manual/index.html>
- Vieira, P. N., Silva, M. N., Mata, J., Coutinho, S. R., Santos, T. C., Sardinha, L. B., & Teixeira, P. J. (2013). Correlates of health-related quality of life, psychological well-being, and eating self-regulation after successful weight loss maintenance. *Journal of Behavioral Medicine*. <https://doi.org/10.1007/s10865-012-9454-9>
- Villena Martínez, V. (2015). *Análisis comparativo de métodos de calibrado para sensores RGB-D y su influencia en el registro de múltiples vistas*. Retrieved from <https://rua.ua.es/dspace/handle/10045/48745>
- WHO, Organisation, W. H. (2006). Estrategia Mundial sobre Régimen Alimentario, Actividad Física y Salud Marco para el seguimiento y evaluación de la aplicación. In *World Health Organization*. <https://doi.org/10.1080/11026480410034349>
- WHO. (2003). Evidence for action. *World Health Organization*. <https://doi.org/10.1088/0004-637X/697/1/880>
- Winter, R. J. (2014). Agile Software Development: Principles, Patterns, and Practices. *Performance Improvement*. <https://doi.org/10.1002/pfi.21408>
- Xiaohui, T., Xiaoyu, P., Liwen, L., & Qing, X. (2018). Automatic human body feature extraction and personal size measurement. *Journal of Visual Languages and Computing*. <https://doi.org/10.1016/j.jvlc.2018.05.002>

7-REFERENCIAS

8 ANEXOS

8.1 Anexo I Bocetos del sistema de visualización

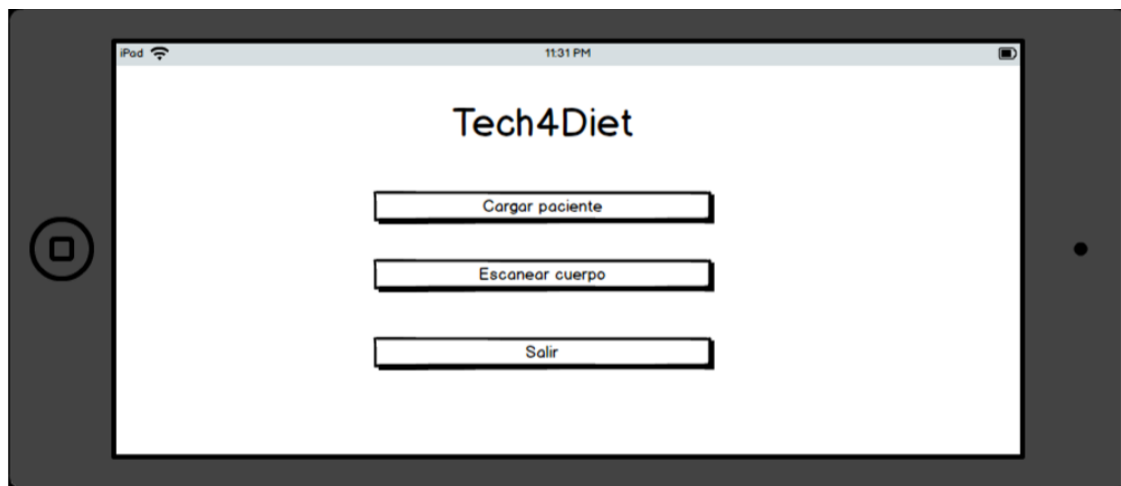


Figura 64 Mockup del menú principal



Figura 65 Mockup de búsqueda de un paciente

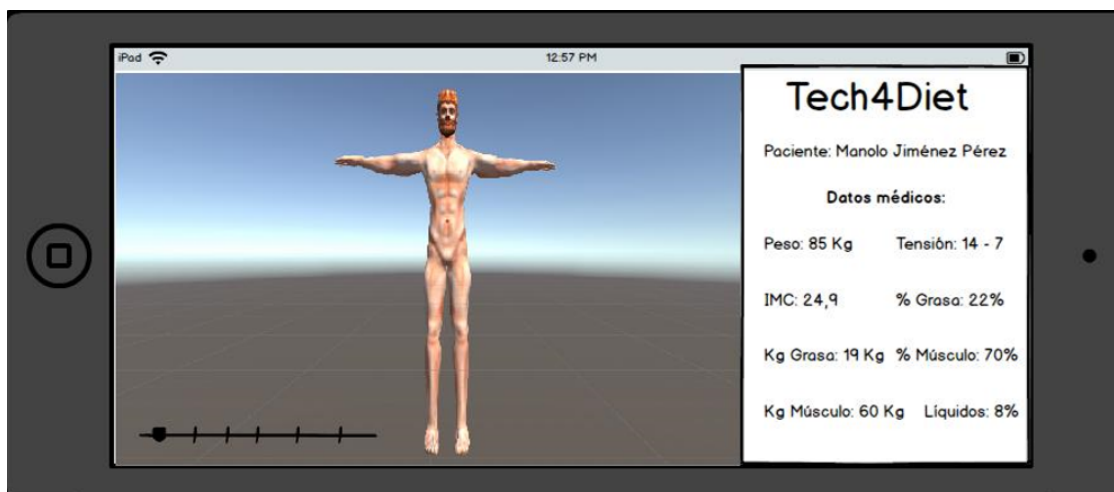


Figura 66 Mockup de la visualización del cuerpo del paciente

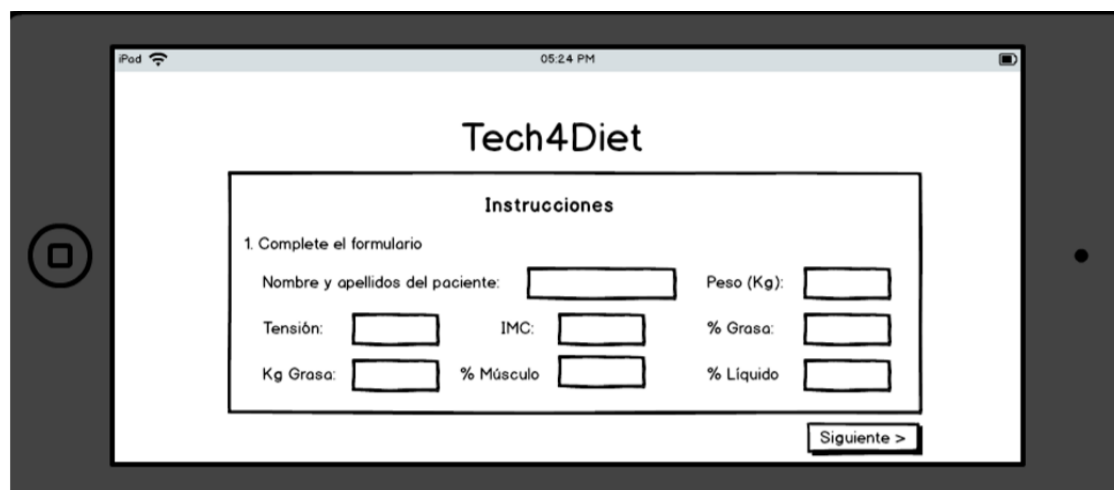


Figura 67 Mockup del formulario sobre el paciente



Figura 68 Mockup de la postura que debe adoptar el paciente

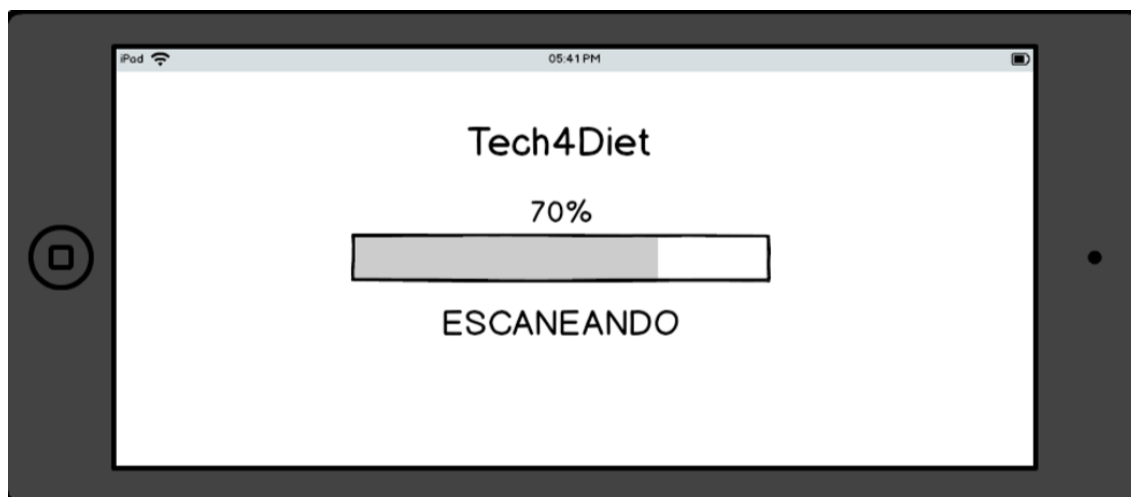


Figura 69 Mockup del progreso de escaneo

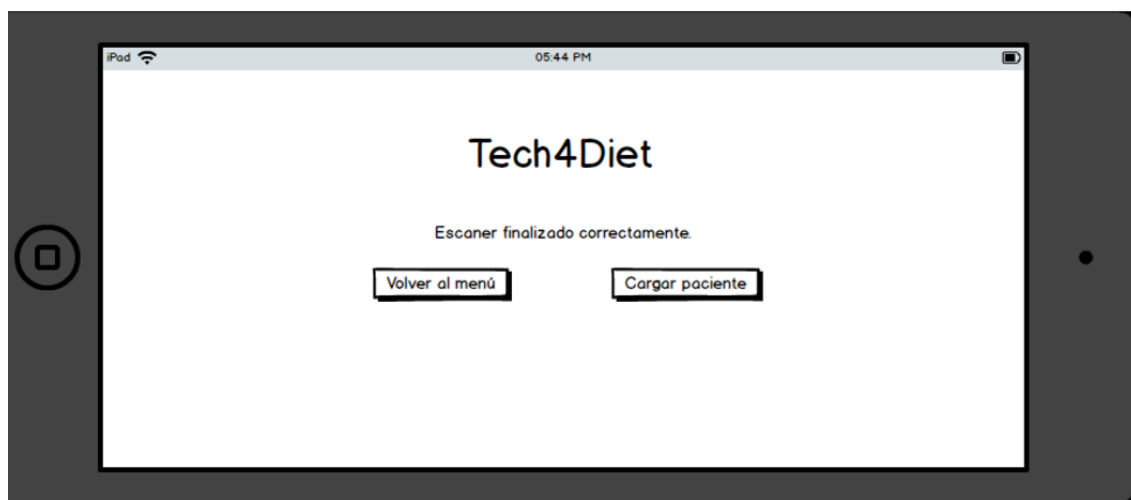


Figura 70 Mockup del resultado del escaneo

8.2 Anexo II Interfaz gráfica

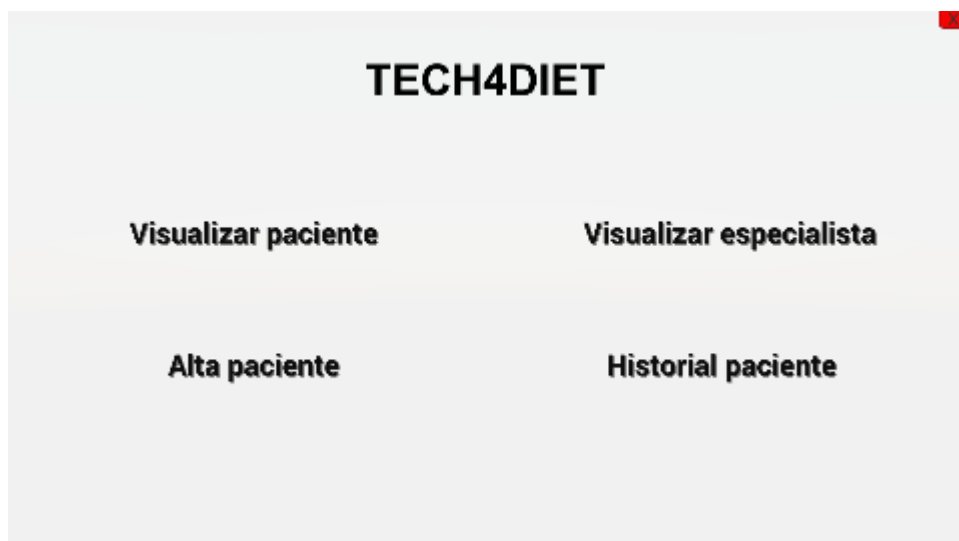


Figura 71 Menú principal



Figura 72 Cargar paciente



Figura 73 Escena especialista

TECH4DIET

Instrucciones

1. Complete el formulario

Nombre	<input style="width: 90%;" type="text" value="Enter text..."/>	Apellidos	<input style="width: 90%;" type="text" value="Enter text..."/>	DNI	<input style="width: 90%;" type="text" value="Enter text..."/>
Sexo (H o V)	<input type="checkbox"/>	Domicilio	<input style="width: 90%;" type="text" value="Enter text..."/>		Población
SS	<input style="width: 90%;" type="text" value="Enter text..."/>	Teléfono	<input style="width: 90%;" type="text" value="Enter text..."/>		Email

Volver
Siguiente

Figura 74 Alta paciente

TECH4DIET

Instrucciones

2. Introduzca a paciente de la misma forma que aparece en la imagen:



VolverEscanear

Figura 75 Asistente

TECH4DIET

Datos administrativosSesionesNueva sesión

Nombre <input style="width: 90%;" type="text" value="Enter text..."/>	Apellidos <input style="width: 90%;" type="text" value="Enter text..."/>	DNI <input style="width: 90%;" type="text" value="Enter text..."/>
Sexo <div>Hombre ▾</div>	Domicilio <input style="width: 90%;" type="text" value="Enter text..."/>	Localización <input style="width: 90%;" type="text" value="Enter text..."/>
SS <input style="width: 90%;" type="text" value="Enter text..."/>	Teléfono <input style="width: 90%;" type="text" value="Enter text..."/>	Email <input style="width: 90%;" type="text" value="Enter text..."/>

ModificarGuardar

Figura 76 Datos administrativos

TECH4DIET

Datos administrativos
Sesiones
Nueva sesión

Peso	Altura	IMC	Medida muñeca	Perímetro brazo	Perímetro bicipital	Perímetro antebrazo	Perímetro muslo
Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...

Perímetro gemelar	Perímetro tórax	Cintura	Cadera	IGG	Perímetro abdomen	Composición	% Grasa
Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...

Modificar
Guardar

Figura 77 Sesiones

TECH4DIET

Rellene los datos y pulse "Guardar" para guardar la sesión

Peso	Altura	IMC	Medida muñeca	Perímetro brazo	Perímetro bicipital	Perímetro antebrazo	Perímetro muslo
Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...

Perímetro gemelar	Perímetro tórax	Cintura	Cadera	IGG	Perímetro abdomen	Composición	% Grasa
Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...	Enter text...

Guardar

Figura 78 Nueva sesión



Figura 79 Progreso escaneo



Figura 80 Fin escaneo